MIKROSZÁMÍTÓGÉP MAGAZIN

1985/3

A NEUMANN JÁNOS SZÁMÍTÓGÉP-TUDOMÁNYI TÁRSASÁG LAPJA

Tanuláshoz, munkához,
játékhoz pro primo:

PRIMO

személyi számítógép

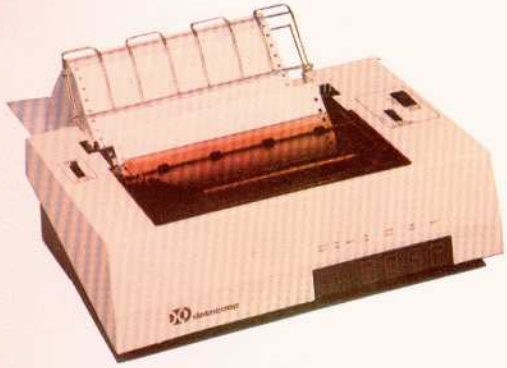
MICROKEY

- BASIC programozási nyelv
- billentyűzet: kapacitív elven működő
  magyar ABC kis- és nagybetűkkel
- megjelenítés: 16 sor × 42 betűhely (szöveg)
  256 × 192 képelem (grafikus)
- program és adattárolás, kazettás magnetofonnal
- TV készülékhez csatlakoztatható
- háromféle változat memória kapacitás (kbájt) szerint:

ROM	RAM	Fogy. ár (Ft)
16	16	11 500,–
16	32	16 150,–
16	48	19 339,–

- saját tápegység (4 600,– Ft)

DCD · PRT · 80
GRAFIKUS MOZAIKNYOMTATÓ



- Nyomtatható karakterek száma soronként: 80
- Nyomtatási sebesség: 80 kar/s
- Karakterkészlet: 95 ASC II/IS
- Kétirányú nyomtatás logikai kereséssel
- Egyidejű másolatok száma: 1 eredeti
  3 másolat
- Mikroprocesszoros vezérlés
- Egykártyás felépítés
- Változtatható papírszélesség
- Grafikus szimbólum készlet
- Bit vezérlésű grafika (tű címzés)
- Traktoros vagy gumigörgős papírtovábbítás
- Papírhiány érzékelés
- Dupla széles karakterek
- Automatikus soremelés
- Festékszalag kazetta
- DZM vagy CENTRONICS interfész

datacoop

Kapható: EMO ELEKTROMODUL

2. sz. bolt
Budapest XIII.
Jászai Mari tér 5.
T: 530-800

MIKROSZÁMÍTÓGÉP MAGAZIN

A Neumann János Számítógéptudományi Társaság lapja

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztő bizottság vezetője:
Kovács Győző

Munkatársak:

Broczkó Péter (hírek)
Buday György István (személyi számítógépek)
Jakab Ágnes (ember–gép kapcsolat)
Kovács Győző (levelezés)
Lindner László (sakkprogramozás)
Pataki Ernő (programozástechnika)
Petróczy Judit (könyvek)
Pogány Csaba (alkalmazástechnika, tanfolyam)
Simonyi Endre (klub)
Takácsy Ildikó (favágás)
Varga András (iskola – számítógép)
Vass Nándor (alkalmazások)
Votisky Zsuzsa (játékprogramok)
Zárda Sarolta (piac)

A szerkesztőség munkatársai:

Albert Tibor
Nacsa Sándor

Felelős szerkesztő:
Könyves Tóth Pál
Szerkesztőség:
1027 Budapest II., Fő u. 68.
Telefon: 154-250

Kiadja: a Delta Szaklapkiadó és Műszaki Szolgáltató Leányvállalat
Felelős kiadó:
Faklen Pál igazgató
1442 Budapest VII., Garay u. 5.
Telefon: 415-583, 215-440

Terjeszti a Magyar Posta
Előfizethető
bármely postahivatalban,
a kézbesítőknél,
a Posta hírlapüzleteiben
és a Posta
Központi Hírlap Irodában
(Budapest V., József nádor tér 1.
Postacím: 1900 Budapest)
közvetlenül
vagy postautalványon,
valamint átutalással
a PKHI 215–96162
pénzforgalmi jelzőszámra.
Előfizetési díj:
egy évre 180,– Ft,
fél évre 90,– Ft.

Szedte:
a Nyomdaipari Fényszedő Üzem
(857482/9)

Nyomás:
Petőfi Nyomda, Kecskemét,
Külső Szegedi út 6.
(85/50063)
Telefon: 28777
Felelős vezető:
Ablaka István igazgató

INDEX: 25629
ISSN 0236–6088

Címképünk:
Magyarországon is bejelentették az IBM PC, PC XT és PC AT típusú személyi számítógépeket

Tartalom

Útijelentés	2
Számítógép a tanulásban	3
COMPUTER–M	9
„A profizmust hiányolom…"	14
Robottechnika	22
Word processing Japánban	24
µprogramok	26
A technológiák szerepe	34
Tudáspróba	47
Adok – veszek – cserélek	47
ISKOLA – SZÁMÍTÓGÉP	
Kísérletek a képernyőn	4
A HT–1080Z számábrázolásáról	5
Gépi kódú kiegészítések BASIC-hez	6
Ahol élnek a lehetőségekkel	7
PROGRAMOZÁSTECHNIKA	
Még egyszer az APL-ről	10
Lokális változók BASIC programokban	11
LOGO – a programozástanulás új útja	13
TANFOLYAM	
Alapozás X.	16
µKLUB	
Építsünk számítógépet! VIII.	29
Mi már építünk!	30
Kazettaszabvány	30
Ki ad magyarázatot?	32
Közérdekű válaszok az olvasóknak	32
SAKKPROGRAMOZÁS	
Specializált számítógépet vagy programot?	36
Lépésről lépésre	38
JÁTÉKPROGRAMOK	
Repülésszimulációk pilóta szemmel	40
AZ OLVASÓ ÍRJA	44
HÍREK, ÉRDEKESSÉGEK	46
KÖNYVEK	48

Útijelentés

„De nem jó, ugy-e nem jó menni a messzi égen,
A felhők rajzain szenvedni oktalan,
És nézni végtelen, némán, boldogtalan
Párist, a legszebb várost, ahol semmim sincs nékem"
(SZÉP ERNŐ: HÁZTETŐ A MONTMARTRE-ON)

Sokáig gondolkoztam, hogy megirhatom-e franciaországi útijelentésemet szerkesztőségi cikk formájában. Az esetek többségében az útijelentés formalitás; nem azért készitik, hogy bárki is elolvassa. Az is zavart, hogy a μM ez évi 1. számában megjelent „nyelvőrködésem" is egy francia–magyar megbeszéléshez kapcsolódott – nem lesz egy kicsit sok a franciákból?

Végül úgy döntöttem, hogy mégis megosztom az élményt az Olvasókkal, tehát elmondom, hogy mit is tettek a franciák a *társadalom informatizálásáért*.

Nem sokkal a látogatás előtt üzleti úton voltam Franciaországban. Szokásom szerint leroskadtam a szállodában, és ugyanazzal a mozdulattal már be is kapcsoltam a televíziót, amelynek képernyőjén rövidesen megjelent két kéz és egy személyi számítógép. Egy barátságos hang éppen azt magyarázta, hogy „ma olyan társadalomban élünk, amelyben mindenkinek ismernie kell a számítástechnikát, tudnia kell kezelni a számítógépeket, mert nézzék csak, ez egyáltalán nem boszorkányság (mutatja), én is megtanultam, mert ha ezt nyomom meg (nyomja), akkor ez van a képernyőn, ha meg azt, akkor meg az. Nem is olyan hosszú idő múlva körülbelül ennyit kell tudnia mindenkinek, a munkásnak és a minisztternek egyaránt."

Nagyon tetszett a műsor. Egy kicsit irigyeltem is az előadót, olyan jól és pergően, sőt meggyőzően mondta a szöveget, és közben mutatta is a gépen, amiről beszélt. Aztán távolodott a kamera, és kiderült, hogy a francia miniszterelnök, Fabius úr volt az előadó.

Ezek után egyáltalán nem csodálkoztam, amikor a mostani tárgyalások alkalmával kiderült, hogy az egy éve elhatározott társadalmi programot miniszterelnöki utasitás gyorsította fel elsősorban azzal, hogy jelentősen megnövelte az anyagi támogatást, és létrehozta azt az intézményrendszert, amelynek a programot végre kell hajtania.

Franciaországban az én ismereteim szerint ez a harmadik olyan központilag támogatott akció, amelynek célja a franciák „fájdalommentes" átvezetése (az egyik tárgyalópartnerem ötletes megfogalmazása) az informatika korába.

Az első volt a hatvanas években a *Plan Calcul*, ami megteremtette Franciaországban a számítástechnikát és az elektronikai ipart. Rengeteg pénzbe került akkor, de később még többe került volna. A második akció 1975-ben a *távbeszélő-hálózat korszerűsítése* és kiterjesztése volt, amelynek eredményeképpen 3 év alatt a 100 főre jutó 5 készülékről 42-re bővült a telefonhálózat, ami azt jelenti, hogy a francia családok 98 százalékának van ma telefonja. Most indult a harmadik akció, amelynek az a célja, hogy *minden francia értsen az informatikához*.

Az előző két vállalkozás alapozta meg ezt a harmadikat, amelyet az első kettő nélkül nem lehetne hatékonyan végrehajtani.

Joggal merül fel a kérdés, hogy szükség van-e egyáltalán a társadalom informatizálására. Nem csupán egy gyorsan terjedő divathullámról van szó, amelyre kidobnak egy csomó pénzt, és az eredmény semmi. Annyit még el kell mondanom – ezt a franciák is többször emlegették –, hogy ami a célkitűzést, a módszereket és az eredményeket illeti, a francia és a magyar társadalmi program nagyon közel áll egymáshoz, majdnem azonosak. Legfeljebb a rendelkezésre álló anyagi eszközökben van vagy két nagyságrendnyi különbség, pedig csak ötször annyian vannak, mint mi. Ezenkívül más volt a célja a program indításának is.

Franciaországban a polgárok megtanítása az informatikára nemcsak gazdasági, hanem egyre inkább politikai kérdés. A francia kormány nehéz időket él át, a szocialista kormányra hárul a feladat, hogy gazdaságtalan és ugyanakkor nagyon sok munkást foglalkoztató iparágakat megszüntessen, bevezesse a robotikát, ami ugyancsak a fizikai munkalehetőségeket csökkenti. Ezeknek az intézkedéseknek a következménye, hogy a munkanélküliek száma sohasem volt olyan magas, mint mostanában.

Nemcsak Franciaországban, de az iparilag fejlett országokban mindenütt minden 10 új munkahelyből csak egy a fizikai. Az elmúlt 10 évben az informatikában alkalmazottak száma ugyanakkor meghatszorozódott. Ezt a franciák világosan látják, és ezért felgyorsították a társadalom informatizálására indított programot. Arra számítanak, hogy az így kiképzett emberekkel feltölthetik a ma még betöltetlen, egymás után keletkező új informatikai munkahelyeket, és ezzel erőteljesen csökkenthetik a munkanélküliek számát. Az sem lényegtelen, hogy ezek az emberek a monoton fizikai munka helyett alkotó szellemi munkával foglalkozhatnak.

Franciaországban – miként nálunk is – az iskola-számítástechnikát is a társadalmi program szolgálatába állították. A kormány „megzsarolta" az iskolákat. Azt mondta, akkor kapnak számítógépeket, ha azokat legalább napi 10 órában használják, elsősorban az iskolai munkában, másodsorban – délután és este – átengedik bárkinek, aki csak bejön az iskolába, hogy dolgozzon rajtuk.

A franciák is ugyanazokkal a problémákkal küzdenek, mint mi, nagyon sokan idegenkednek a gépektől, nem tudják eldönteni, leüljenek-e melléjük vagy sem, vegyenek-e gépet vagy ne. Franciaországban ebbe a kategóriába tartoznak a kisfarmerek, a kereskedők, a magánvállalkozók. Arra biztatják őket, hogy menjenek be a legközelebbi iskolába, üljenek le a gépek mellé és csinálják meg a mérleget, a napi elszámolást, aztán vigyék haza a mágneslemezt, és ezt egészen addig folytassák, amíg nélkülözhetetlennné nem válik számukra a gép. Akkor már bátran vehetnek egyet.

Valahogyan itt hat vissza a társadalmi program a gazdasági életre. Franciaországban ugyanis az amerikai, japán, de még az angol számítógépek ára is jóval alacsonyabb, mint a francia gépeké. Azt hiszem, hogy a társadalmi programra szánt pénzért azért vásároltak végül francia gyártmányú gépeket, mert ezzel nemcsak fel tudták futtatni a számítógépgyártást a gazdaságos sorozatnagyságra, és így versenyképes árakat tudtak produkálni, hanem azért is, mert akik az iskolai számítóközpontokban a francia gépeket ismerik meg, azok később ezeknek a vásárlóivá válnak. Így a gazdasági vezetés arra számíthat, hogy a gyártási volumen még tovább növekszik, ami újabb munkaalkalmakat jelent.

A rendszeres felnőttoktatásra és a társadalom számára hasznos projektek támogatására a kormány önálló intézményrendszert szervezett, az ADI-t (Agence de l'Informatique) és az általa irányított klubmozgalmat, amelyet X 2000-nek neveztek el.

Az ADI feladata, hogy állami pénzekből támogatást adjon olyan projekteknek, amelyek az informatika társadalmi méretű elterjedését segítik. Csak néhány példa. A francia posta létre akarta hozni az elektronizált postahivatalokat. Az ADI azt mondta: ez egy előremutató projekt, támogatom. Megbeszélték, hogy vásárolnak jó néhány gépet, és a projektben részt vevő postahivataloknak szétosztják, mindenki csináljon vele, amit tud. Volt olyan hivatal, amelyik köszönettel nem kért a gépekből, de volt, amelyik több gépet is kért. Megkapta. Aztán egy-két hónap múlva az ADI megnézte, hogy mit csináltak a gépekkel. Ahol okos dolgokat találtak ki, ott újabb támogatást kaptak, például programozói segítséget vagy újabb gépeket. Egy fél év alatt elkészült vagy félszáz alkalmazói program. Ezeket összefésülték, megigazították, rendszerbe fogták össze, és még azon az őszön bemutatták a SICOB-on az elektronikus postahivatalt. A programokat sokszorosították, megszervezték a többi postahivatalban a tisztviselők betanítását, és a probléma el volt intézve. Egyébként elmondták, hogy a szakértőknek az volt a véleményük, hogy egy rendszert nem szabad így létrehozni. Ők a szokásos utat javasolták: tanulmány, elfogadják, módosítják, véglegesítik, megtervezik, programozzák, tesztelik, bemutatják, elfogadják, bevezetik – és erre 2–3 évet szántak. A postásoknak azonban jobban tetszett a fél év – szabálytalanul. Ki tudja, miért?

Ugyanígy segítette az ADI a minisztériumok információs rendszerének kialakítását. Külön felelőse van a fizikailag hátrányos helyzetű gyerekek informatikai oktatásának, támogatják a sport és az informatika közötti együttműködést (például az edzők továbbképzése terminálon!), de a turisztikai és utazási adathálózat kialakítását és más, fantasztikusnak tűnő projekteket is.

Az X 2000 és a magyar mikroklubok között megint csak az a különbség, hogy ők sokkal gazdagabbak. Az X 2000 klubokat az X 2000 Fondation hozza létre. Ha valahol egy ilyen klubot akarnak szervezni, akkor elsőrendű követelmény, hogy a helyi szervezeteknek pénzük legyen, mert az X 2000 Fondation az alapításhoz csak annyival járul hozzá, amennyivel a helyi közigazgatás, gyárak és intézmények együttesen. A Fondation gépeket és oktatási technológiát ad, gyakorlatilag minden segítséget – kiképezi például a személyzetet –, hogy a klub jól működjön.

Az anyagi támogatás csak az induláshoz elég, aztán a központi támogatás megszűnik, a klubnak el kell tartania magát. Ezért üzleteket kötnek, például a környék gyáraival, amelyeknek munkásait a gyár költségén kiképezik és továbbképzik. Aki munkanélküli, annak nem kell a tanfolyamokon fizetnie, és ingyen használhatja az X 2000 klub valamennyi gépét és a könyvtárat. Ha akar, beülhet melegedni is. Ha egyszer odamegy, akkor ott is marad, erre számítanak. A gép- és könyvtárhasználat a tanfolyami időn kívül mindenkinek ingyenes. Az X 2000 klubok évente nagy összejöveteleket tartanak, amelyek célja a tapasztalatok kicserélése, új oktatási módszerek bemutatása, egyáltalán az évi egyszeri találkozás a személyi kapcsolatok kialakítására.

Megnéztük az ASPASIE klubot Torcyban. Torcy egy kisváros kb. 60 km-re Párizstól, a Val Maubuée új városnak a központja. Franciaországban is van ugyanis olyan program, amely a kisebb településeket egyesíti, ezáltal új városok jönnek létre. Val Maubuée is így született, hat kis településből, amelyek megtartották önállóságukat, sőt a nevüket is. Az új város elnökét választják. Val Maubuée vezetője Torcy polgármestere. Vannak dolgok, amelyeket közösen intéznek, közösen építenek. Ilyen volt az ASPASIE Központ is, ami ha úgy tetszik, akkor információs központ, de kultúrház is, klub is. („Sokba került, de megérte" – mondta az elnök.)

Van ennek az ASPASIE-nek egy különlegessége: saját információs rendszert üzemeltet. A rendszer megértéséhez el kell mondani, hogy a már említett telefonhálózat korszerűsítéséhez a népszerűen elektronikus telefonkönyvnek nevezett rendszer is hozzá tartozik. Franciaországban minden évben adnak ki telefonkönyvet, amelyről a megbeszélés résztvevői egymást túllicitálva közölték, hogy „boszszantó, mert a sok változás miatt az év végére már gyakorlatilag használhatatlan, ezért a kinyomtatása kidobott pénz".

A posta is így gondolta, ezért határoztak úgy, hogy a mintegy 22 millió telefonkészülék mellé ugyanannyi Minitel terminált is elhelyeznek – természetesen ingyen – a telefonkönyv helyett. A terminállal egész Franciaország összes telefonszámát le lehet kérdezni. Nem akartam elhinni, de azt mondták, a telefonkönyv költségének, valamint a szerkesztési, oda- és visszaszállítási költségnek a megtakarításával a terminál két-három év alatt amortizálódik. A telefonelőfizetők egy része a terminált nem igényli, ők egyelőre telefonkönyvet kapnak, a többiek előre meghatározott terv szerinti időben a terminált kapják meg. Val Maubuée vezetése ezekre a terminálokra építette a tervét és alakította ki az új város információs rendszerét.

A posta hatékony közreműködésével létrehoztak egy kis lokális hálózatot, amelynek használata természetesen ingyenes. A központi gép az ASPASIE-ben van, a csatlakozó adatbázisok pedig az egyes településeken. Minden számítógép kétféle célt szolgál. Egyrészt ezekben tárolják a mindenkit érdeklő adatokat (a sport- és kulturális eseményeket, a polgármester közleményeit, az országos rendezvények idejét stb.), másrészt postaládaként használják. Ebben bárki elhelyezhet címzett vagy mindenkinek szóló közleményeket, üzeneteket.

Miután minden lakásban van telefon, a postaládától azt várják, hogy közelebb hozza egymáshoz az új város lakóit, például az azonos érdeklődésűek megtalálják egymást, de közelebb kerül egymáshoz a városok vezetése és a polgárok is. A polgármester maga is arra számít, hogy az emberek nem járnak be a városházára ügyeiket elintézni, hanem leadják azokat a postaládába, aztán megkapják a választ, esetleg vitatkoznak is. Úgy számolta, hogy kb. napi két órát fog a terminálnál az üzenetek megválaszolásával eltölteni, és nem 4–5 órát félfogadással.

A rendszert tovább akarják fejleszteni. Ha már mindenki hozzászokott a terminálhoz, akkor lehetőség lesz például otthoni munkahelyek kialakítására, hiszen a lokális hálózatokon keresztül elérhető lesz a postai adathálózat is, akkor pedig a munkahelyi számítógépeken is lehet hazulról dolgozni.

Lehetne még beszélni az iskolai számítógép-hálózatról, ahol a diák meg tudja tanulni az adatkezelést és megismerheti a telekommunikációs lehetőségeket, a hátrányos helyzetű gyerekek részére készített speciális számítógépekről és perifériákról, az óvodás korú gyerekek játékos programozástanításáról, de be kell fejeznem.

Amit láttunk, az nálunk sem elérhetetlen; illetve csak azt a részét nem tudjuk megvalósítani – sajnos –, amihez az infrastruktúra hiányzik, az olcsó számítógép és a telefonhálózat.

Azt hiszem, ha egy iparilag fejlett országban érdemes az informatika társadalmi méretű bevezetését támogatni, akkor ez nálunk sem kidobott pénz – igaz, a haszon nem jelentkezik közvetlenül és azonnal. De jelentkezik!

KOVÁCS GYŐZŐ

Számítógép a tanulásban

A számítógép az oktatásban nagyon mély és radikális változást fog eredményezni. A változás befolyásolásához új adatokra, statisztikákra és elképzelésekre van szükségünk. Elkerülhetetlen a gondolkodás új elméleti megfogalmazása.

A hagyományos oktatási pszichológia bizonyos beavatkozások hatását vizsgálja. Például: mi a hatása a halmazelmélet tanulásának a számolási képességre? A számítógép esetében a gép hatását vizsgálják a gondolkodás fejlődésére, az iskolára, az osztályra. Ez azonban teljesen értelmetlen. Azt feltételezik, hogy a programozás kognitív hatása jelentkezik a megjavult tervezési képességben.

Ha a programozás befolyásolja a gondolkodást, akkor ez közvetlenül nem a programozás elsajátítását jelenti. A hatása inkább indirekt. A programozási jártasság lehetővé teszi, hogy a tanár új módon tanítson, a diákok pedig új módon tanuljanak.

A számítógép olyan radikális változást eredményez a tudáselsajátítás folyamatában, mint amilyet a ceruza használata hozott az oktatásban. Ennek kezdeteit már látjuk a számítógép széles körű felhasználásában: egyesek számára már ma is napi használati eszköz.

Mostanában érkeztünk el oda, hogy lehetségessé válik a számítógép oktatásban történő alkalmazásának alaposabb megfigyelése. Ennek előfeltétele, hogy viszonylag sok számítógép álljon rendelkezésre. Remélhetőleg közeleg az az idő, amikor minden tanulónak lesz számítógépe.

Ma még súlyos helyzetet teremt a számítógéphiány. A gépek jelenlegi használata inkább kárt okoz az oktatás szempontjából, mert aláássa a képzelőerőt és erősíti a téves felfogásokat. A legtöbb fiatal nem kerül a géppel olyan kapcsolatba, hogy ellenőrizni tudná azt. Ez a hiányosság persze nem teszi szükségtelenné a számítógépek használatát, de fontos a számítógép és ember viszonyának megfogalmazása.

Kísérletek bizonyítják, hogy az iskolában inaktív gyerekek a LOGO használata közben igen aktívakká váltak. A program használatával törekedtek eredményeik javítására, és végül elemi módon kapcsolatba kerültek a matematikával. A LOGO felhasználása három változó: a sebesség, az idő és a távolság megismerését tette lehetővé. Nem valószínű, hogy ezt számítógép nélkül el lehetett volna érni. A példából látszik, hogy a számítógép segítségével olyan személyiségtípusok is bekapcsolhatók az oktatásba, tanulásba, akik a hagyományos oktatásban nem motiváltak. A gép módot nyújt arra, hogy a tanulás tárgyát és módját a személyre szabjuk.

A számítógép alkalmazásával megvalósítható az ún. tanítás nélküli tanulás: kissé eltúlozva, a gép használatával a gyerekek egyedül tudnak tanulni. Ma már vannak tapasztalatok olyan iskolákból, ahol sok a számítógép. Itt a tanár szerepe megváltozott: már nem oktat, hanem figyeli a diákok munkáját, segít, ha valaki elakad.

A számítógépnek megvan az a különleges tulajdonsága, hogy interaktív és kísérleti. Így a tudás elsajátítása lépésenkénti és folyamatos lehet. A LOGO létrehozásának is ez volt a fő gondolata. Jelentőségét akkor értjük meg igazán, ha szembeállítjuk a tanulás és tanítás eme új felfogását a hagyományossal, ahol az ismeretátadás és elsajátítás mindig egyszerre, nagyobb adagokban történik, passzív befogadás mellett.

●

Írásunk *Seymourt Papert* New Theories for New Learnings című, a School Psychology Review-ben 1984 októberében megjelent publikációja alapján készült.

– illyés –

Kísérletek a képernyőn

Első osztályban és a második osztály első hónapjaiban, amíg a technika tantárgy keretében sorra nem kerülnek a számítástechnikai ismeretek, csak azok a tanulók tudják használni a számítógépet, akik általános iskolában vagy a gimnáziumban részt vettek számítástechnikai szakköri munkában. Kiemelkednek azok a diákok – és egyre többen vannak –, akiknél már otthon is működik egy-egy ZX81, Spectrum vagy Commodore 64.

Az alábbiakban arról lesz szó, hogyan használható a számítógép a kémia- és fizikaórákon a gimnáziumokban.

Mindenesetre a számítógép akkor alkalmazható jól az oktatásban, ha a tanulók hozzászoknak a jelenlétéhez, és nemcsak kivételes esetekben találkoznak vele. Ezért mindegyikük számára lehetőséget kell teremtenünk az egyéni ismerkedésre a géppel.

A TII-től kémiai és fizikai témákat feldolgozó oktatási programokat kaptam kipróbálásra, a pedagógiai gyakorlatban való alkalmazhatóság vizsgálatára. A programok két gépen, a HT–1080Z és az ABC–80 számítógépen futtathatók.

A gimnáziumok első osztályos fizika tananyagához a TII programcsomagot forgalmaz, amely a munkafüzet 6., 7. és 8. feladatlapjához kapcsolódik. A programcsomagnak az a legnagyobb előnye, hogy a tanulók kísérleteit kiegészíti, illetve gyorsabbá teszi, de az értelmezést a tanárra bízza.

A 6. feladatlap a tankönyv „Miért terjed ki a gáz?" című témájához kapcsolódik. Célom itt az volt, hogy a tanulókkal elsajátíttassam a termodinamika második fő tételét, mint a statisztikus törvényszerűségek eredményét. Felhívtam figyelmüket a modellfogalom lényegére, hogy az bemutatja a tanulmányozott jelenség bizonyos jegyeit, szerkezetét és dinamikáját.

Az óra elején összefoglaltuk a gázok addig megismert tulajdonságait és az ezek alapján kialakított modellképünket. Ezután egy játékos modellkísérlettel haladtunk tovább, melyet a tanulók párokban végeztek el. A kísérlet címe „Játék a darazsakkal". Lényege a következő.

Játék a darazsakkal

Minden pár két egymásba nyíló szobát rajzol. Az egyik szobába 6 darab, 1–6-ig megszámozott papírdarabkát tesznek, a játékosság kedvéért ezeket darazsaknak nevezzük. A darazsak véletlenszerűen röpködnek az egyik szobából a másikba, amit a tanulók először dobókocka segítségével biztosítanak. Amelyik darázs számát dobják, az átrepül a másik szobába. Minden tizedik dobás után fel kell jegyezniük a darazsak eloszlását a szobákban.

A kísérlet kiértékelésekor kiderül, hogy a leggyakoribb eloszlás az, amikor három darázs az egyik és három a másik szobában van, és a legkisebb valószínűséggel fordul elő az, hogy mind a hat darázs az egyik helyiségben lenne.

Sok dobás elvégzésére nincs idő, ezért a tanulók elkezdik ugyan a kísérletet, de húsz-harminc dobás után abbahagyják, és a programot tekintjük meg.

Elmondom, hogy ez egy szimulációs program: a képernyőn figyelemmel követhetjük, hogy melyik darázs mikor megy át. A kockadobás szimulációjára a számítógépbe épített véletlenszám-generátor szolgál, amely ebben az esetben 1 és 6 között állít elő számokat. 60 dobás után a gép kiírja minden darázs átrepülése utáni eloszlást, így a folyamatot dinamizmusában lehet látni.

A program segítségével a kísérletet sokkal gyorsabban tudjuk elvégezni. A darazsak eloszlásának kiírása után a bal illetve a jobb oldalon átlagosan tartózkodó darazsak számát grafikon ábrázolja. Így igen szemléletesen látható az egyenletes eloszlás.

A darazsas játék modellszerűen magyarázza a gázok szétterjedését, és ennek felhasználásával jutunk el a termodinamika második fő tételének egyik megfogalmazásához.

Osztozkodás az energián – Az energia egyenletes eloszlása

A 7. és 8. feladatlaphoz kapcsolódó programok is a kockadobások felgyorsítását teszik lehetővé, és így alapvető fizikai törvények statisztikus értelmezéséhez adnak segítséget.

Disszociáció

Ezzel a programmal egy elemi kémiai reakciót figyelhetünk meg, amelyben három anyag vesz részt. Megvizsgálhatjuk az A + B → C és a C → A + B reakciótípusokat, valamint mindkét irányú reakciót egyszerre. Megadható mindkét reakció sebessége, és az egyes anyagok kiindulási koncentrációja. A program kétféle eredményt adhat: vagy az egyes molekulák százalékos megoszlásáról készít grafikont, vagy a reakciótérben zajló eseményeket figyelhetjük meg. A bemenő paraméterektől függően megfigyelhető az egyesülés, a bomlás és a disszociáció.

A periódusos rendszer elemei

Ezt az oktatóprogramot tartom a leghasznosabbnak és legeredményesebbnek. Játékos formában gyakoroltatja az elemek nevének és vegyjelének, a periódusos rendszerben elfoglalt helyüknek és annak az elektronszerkezettel való kapcsolatának ismeretét. A programnak három üzemmódja és három nehézségi fokozata van: 1. a megjelenő vegyjelekhez tartozó neveket kell beírni; 2. a megjelenő elemek vegyjelét a nyilak segítségével a periódusos rendszerben elfoglalt helyüknek megfelelő négyzetbe kell mozgatni; 3. a véletlenszerűen kiválasztott elem elektronszerkezetét kell beírni.

Minden játék végén, 10 kérdés után a program értékel, a hibás válaszokat kiemeli, és a periódusos rendszerben villogtatja ezeknek az elemeknek a vegyjelét. A képernyőn megjeleníthető a teljes periódusos rendszer. A kérdésekre adott válaszokat hangeffektusok kísérik, ami motiváló hatású.

ELEK

Segítségével az elemek elektronszerkezetét lehet gyakorolni. Két üzemmódban használható. Az első esetben megjeleníti egy elem elektronszerkezetét. A második üzemmódban egymás után néhány elem elektronszerkezete jelenik meg, esetleg hibásan, és el kell dönteni időre (több sebességi fokozat van), hogy helyesek-e. A játékot a gép pontozza.

OKTE

A molekulák elektronszerkezetének gyakorlását teszi lehetővé. Szintén két üzemmódja van. Az elsőben megjeleníti egy molekula elektronszerkezetét. A másodikban a molekulák hiányosan jelennek meg a képernyőn, és a tanulónak kell kiegészíteni azokat.

Sajnos a program csak 19 molekula elektronszerkezetét képes ábrázolni, így a választási lehetőség kicsi. A játék végén a számítógép összesítést végez, és értékeli a tanuló munkáját.

Analitikai program

Ugyanúgy, mint az előző kettő, a számítógépnek azt a tulajdonságát használja ki, hogy nagy mennyiségű adat tárolására képes. Ezzel a programmal 15 fajta ionos vegyület reakcióit lehet tanulmányozni 9 fajta reagenssel. Minden párosításnál kiírja a számítógép, hogy van-e változás a reakció során, vagy nincs, illetve a keletkezett oldat vagy csapadék milyen színű. A tanuló választhat egy olyan üzemmódot, amelyben egy ismeretlen vegyületet kell meghatározni reakciói alapján. A számítógép 15 vegyület közül választja ki véletlenszerűen az ismeretlent, ezek közül 7 kationja a Freseniusféle csoportosítás szerint a kationok első osztályába, a maradék 8 a harmadik osztályába tartozik.

MEND

Az elemek tulajdonságainak változását a periódusos rendszer egy kiválasztott oszlopán (VIIIa) belül lehet megfigyelni, gyakorolni ezzel a programmal, amely tulajdonképpen egy „jóslás-játék". A program a VIIIa oszlopban különböző fizikai sajátságokat tüntet fel, de egy érték a kiírásból mindig hiányzik. Ezt kell a

tanulónak megjósolnia. A program célja, hogy megismertesse a periódusos rendszer egy oszlopán belül a fizikai tulajdonságok változásának tendenciáját.

Kémiai egyensúly

A program a $H_2 + I_2 \rightleftarrows 2HI$ egyensúlyi folyamat dinamikájának demonstrálására készült. A hidrogén-jodid képződése egyike ama kevés folyamatnak, amelyek mindkét irányban egyszerű mechanizmus szerint, jól mérhető sebességgel mennek végbe. A programban megadhatók a kezdeti koncentrációk, majd a képernyőn egyszerre jelenik meg numerikusan és grafikusan a kiindulási és az egyensúly beállta utáni összetétel százalékosan. Az egyensúlyi keverékben változtatható valamelyik anyag mennyisége. Ha az egyensúlyban levő rendszer valamelyik anyagának koncentrációját megváltoztatjuk, akkor átalakulás után új egyensúly alakul ki. Ezt látványosan és gyorsan szimulálja a program.

A Millikan-kísérlet szimulációja

Az első osztályos kémiához, illetve a negyedikes fizikához kapcsolódik. A Millikan-kísérlet célja az elektron töltésének meghatározása. Töltéssel rendelkező olajcsepp esik a gravitáció következtében két elektróda között. A tanuló feladata, hogy a lemezek között olyan előjelű és akkora feszültséget létesítsen, hogy a csepp lebegjen. A feszültségérték szabályozása az olajcsepp felfelé, illetve lefelé irányuló elmozdulása alapján történik. A képernyőn a feszültség és az olajcsepp sebessége jelenik meg.

Lebegés esetén a „kísérleti" üzemmódból ki lehet lépni, de a szimulált kísérlet többször megismételhető. A tanuló feladata, hogy ha adott csepptömeg mellett $1{,}6 \cdot 10^{16}$ V feszültség állítana meg egy -1 coulomb töltésű cseppet, akkor mondja meg, hogy hány coulomb volt az egyes cseppek töltése, és mennyi ezek legnagyobb közös osztója.

A program a választ értékeli, felhívja a figyelmet az esetleges előjeltévesztésre, vagy hibás érték esetén a számítási hibára, arra, hogy a tanuló esetleg kevés kísérletet végzett, végül kiírja az elektron töltésének helyes értékét.

A program az olajcsepp tömegét véletlen szimulálja, töltése csak az elemi töltés egész számú többszöröse lehet. A program ezekhez az értékekhez határozza meg a lebegéshez szükséges feszültséget, melyet a tanulónak az olajcsepp mozgása, illetve lebegése alapján kell meghatároznia. A megadott adatot a számítógép összehasonlítja a tárolt, már kiszámított értékkel, egyezés esetén dicsér, egyébként felhívja a figyelmet a hibára.

A jövőben egyre nagyobb „tudású" és memóriájú számítógépek fognak az oktatásba kerülni. A finomgrafikával rendelkező gépek és a több szín használatával, a lemezgységek és mérőműszerek csatlakoztatásával megváltozik a számítógépek alkalmazási lehetősége, és oktatásban is emberközelibb programok készülhetnek.

SIEGLER GÁBOR

A HT-1080Z számábrázolásáról

Osszuk el a 7-et 3-mal a számítógép segítségével, azaz adjuk ki a PRINT 7/3 parancsot! A képernyőn megjelenő eredmény: 2.33333. Ha most a következő parancsot adjuk a gépnek: PRINT 7/3–2.33333, akkor a képernyőn megjelenő

3.Ø9944E–Ø6 (= $3.09944 \cdot 10^{-6} = 0.0000030944$)

szám azt mutatja, hogy *a számítógép által elvégzett művelet eredménye és a képernyőn megjelenő szám általában nem egyenlő egymással.*

Nézzünk még néhány további, meglepőnek tűnő példát! Adjuk a következő parancsokat a számítógépnek:

```
A = 7 : B = 3 : PRINTA/B
      (az eredmény: 2.33333)
A# = 7 : B# = 3 : PRINTA#/B#
      (az eredmény: 2.3333333333333)
C# = 7/3 : PRINTC#
      (az eredmény: 2.333333253860474)
PRINT 1ØØØØØ/Ø.ØØØØ1
      (az eredmény: 1E+1Ø)
A = 1ØØØØØ/Ø.ØØØØ1 : PRINTA
      (az eredmény: 1E+1Ø)
A = 1ØØØØØ/Ø.ØØØØ1 : PRINT A–1E+1Ø
      (az eredmény: 1Ø24)
A# = 1ØØØØØ/Ø.ØØØØ1 : PRINTA#
      (az eredmény: 1ØØØØØØ1Ø24)
```

Ezek a számítások arra is rámutatnak, hogy a *különböző típusú változók használata módosíthatja a képernyőn megjelenő eredményt.* Az így szerzett tapasztalatok megértéséhez meg kell ismerkedni az iskolaszámítógép belső számábrázolásával. Az egyszeres és a duplapontos számábrázolást fogjuk vizsgálni.

Ehhez nagy segítséget nyújt a következő program:

```
1Ø CLS
2Ø A = 1 : A# = 1
3Ø I = VARPTR(A) : J = VARPTR(A#)
4Ø FOR L = I–3 TO I+3 : PRINT PEEK(L);
   : NEXTL : PRINT : PRINT
5Ø FOR L = J–3 TO J+7 : PRINT PEEK(L);
   :NEXTL
6Ø GOTO 6Ø
```

A 3Ø-as sorban szereplő VARPTR függvény azt a memóriacímet adja meg, amelyiken az A (illetve az A#) változó értékének az ábrázolása kezdődik. A PEEK függvény segítségével pedig adott című memóriabájtok tartalmát tízes számrendszerbeli alakjukban tudjuk kiíratni a képernyőre.

A programot futtassuk le! Az eredmény a következő lesz:

```
 1.  2.  3.  4.  5.  6.   7.  8.  9.  10.  11.
 4   Ø   65  Ø   Ø   Ø   129
 8   Ø   65  Ø   Ø   Ø    Ø   Ø   Ø   Ø   129
```

Az A, illetve az A# változónév helyett más változónévvel (például B és B#, X és X#, A1 és A1#, CD és CD#, CD1 és CD1#) futtatva a programot, mindig sorban csak a 2. és a 3. bájtok értékei változnak.

Ha a változónév egy karakterből áll, akkor a 2. bájt értéke Ø, a 3. bájt értéke pedig a karakter ASCII-kódja. Ha a változónév két karakterből áll, akkor a 2. bájt értéke a 2. karakter ASCII-kódja, a 3. bájt értéke pedig az 1. karakteré. Innen is látható, hogy ha a változónév két karakternél hosszabb, a számítógép akkor is csak az első két karaktert veszi figyelembe.

Változtassuk most az eredeti programban az A, illetve az A# változók tartalmát, és futtassuk így is a programot. Megfigyelhetjük, hogy az eddigi futtatások során az eredmény 1. oszlopában álló két szám nem változott. Ez azért van így, mert mindkét sorban az 1. bájt értéke a változó típusát mutatja. Ez az érték egyszeres pontosságú változó esetén 4, ami azt is jelenti, hogy a változó értékét a nevet követő 4 bájton ábrázolja.

Dupla pontosságú változó esetén az 1. bájt értéke 8, és a változó értékét 8 bájton ábrázolja a gép. A továbbiakban az utóbb említett 4, illetve 8 bájt jelentésével fogunk foglalkozni.

Futtassuk a következő programot:

```
1Ø CLS
2Ø INPUT A
3Ø I = VARPTR(A)
4Ø FOR L = I–3 TO I+3 : PRINT PEEK(L);
   : NEXTL : PRINT : PRINT
5Ø GOTO 2Ø
```

Adjuk az A változónak 2 különböző egész kitevőjű hatványértékeit nullától növekvő, majd csökkenő sorrendben (1,2,4,8,16,... ...,1,Ø.5,Ø.25,Ø.125,Ø.ØØ625,... stb.). Látható, hogy ha a hatványkitevőt 1-gyel növeljük, akkor a 7. bájton 1-gyel nagyobb, ha a hatványkitevőt 1-gyel csökkentjük, akkor a 7. bájton 1-gyel kisebb szám jelenik meg. Ha most ugyanezeknek a számoknak a -1-szeresét adjuk meg, akkor az eltérés az előző eredményektől csupán annyi, hogy a 6. bájton megjelenik a 128 ($= 2^7$) érték.

Adjuk az A változónak a $2^n + 2^{n-1}$ (n egész) alakú értékeket (például: $2^1 + 2^0 = 3$, $2^2 + 2^1 = 6$, $2^7 + 2^6 = 192$, $2^{-1} + 2^{-2} = 0.75$,... stb.). Látható, hogy a 6. bájt értéke mindig 64 ($= 2^6$).

Adjuk az A változónak a $2^n + 2^{n-2}$ (n egész) alakú értékeket (például: $2^2 + 2^1 = 5$, $2^7 + 2^5 = 160$, $2^{-1} + 2^{-3} = 0.625$,... stb.). A 6. bájt értéke most mindig 32 ($= 2^5$).

Végül adjuk az A változónak a $2^n + 2^{n-8}$ (n egész) alakú értékeket (például: $2^8 + 2^0 = 257$, $2^{10} + 2^2 = 1028$, $2^5 + 2^{-3} = 32.125$,... stb.). Ezeknél a példáknál az 5. bájt értéke lesz mindig 128 ($= 2^7$).

Ha most ugyanezen értékek -1-szereseivel is lefuttatjuk ezt a programot, akkor csak annyi lesz a változás az eredményekben, hogy a 6. bájt

értéke az előbbinél 128-cal nagyobb lesz.

Megfigyeléseink magyarázata abban rejlik, hogy a HT-iskolaszámítógép az egyszeres pontosságú valós típusú változók értékeit *kettes számrendszerben* ábrázolja. Ehhez a Ø-tól különböző számokhoz a következő, ún. normálalakot használja:

$m \cdot 2^k$, ahol $1/2 \leq |m| < 1$, a k pedig egész szám.

A k az ún. karakterisztika, értékét 128-cal megnövelve a 7. bájton láthatjuk. Az m mantisszát 3 bájton (6., 5., 4.), (azaz $3 \cdot 8 = 24$ biten), *tehát 24 kettes számrendszerbeli számjeggyel ábrázolja.*

Az m mantisszára megadott feltétel miatt annak első kettedes jegye mindig 1. Ezt használja fel a számítógép arra, hogy a szám előjelét jelölje. Pozitív szám esetében ezt a számjegyet Ø-ra változtatja, negatív számnál pedig meghagyja 1-esnek. (Természetesen ha a gép egy pozitív számmal műveletet végez, akkor ezt a bitet helyreállítja.) *A Ø számot úgy ábrázolja a számítógép, hogy a 7. bájt tartalma Ø.*

Tanulmányozzuk az elmondottakat a következő példákon:

$3 = 11_2 = Ø.11_2 \cdot 2^3$

– a karakterisztika: 3
a megnövelt karakterisztika: $131 = 3 + 128$
a 7. bájt tartalma: 10000011_2 (= 131)
– a mantissza: $Ø.11_2$
a kettedes pont utáni 24 számjegye:
11000000 00000000 00000000
a szám pozitív, ezért az első kettedesjegy értékét Ø-ra változtatjuk, és így
a 6. bájt tartalma: 01000000_2 (= 64)
az 5. bájt tartalma: 00000000_2 (= Ø)
a 4. bájt tartalma: 00000000_2 (= Ø)

$-3 = -11_2 = -Ø.11_2 \cdot 2^3$

– a karakterisztika ugyanaz, mint az előző példában: 3
a 7. bájt tartalma nem változott
– a mantissza: $-Ø.11_2$
a kettedes pont utáni 24 számjegy nem változott és a szám negatív, ezért
a 6. bájt tartalma: 11000000_2 (= 192)
az 5. és a 4. bájt tartalma nem változott.

$Ø.1 = Ø.\overline{1100}_2 \cdot 2^{-3}$

– a karakterisztika: -3
a megnövelt karakterisztika: $125 = -3 + 128$
a 7. bájt tartalma: 01111100_2 (= 125)
– a mantissza végtelen szakaszos kettedes tört: $Ø.\overline{1100}_2$
a kettedes pont utáni 24 számjegye:
11001100 11001100 11001100
a szám pozitív, ezért az első kettedesjegy értékét Ø-ra állítjuk, továbbá, mivel a kettedes pont utáni 25. számjegy 1, ezért felfelé kerekítünk és a 24. számjegyet -1-re állítjuk, így
a 6. bájt tartalma: 01001100_2 (= 76)
az 5. bájt tartalma: 11001100_2 (= 204)
a 4. bájt tartalma: 11001101_2 (= 205)

$-Ø.1 = -Ø.\overline{1100}_2 \cdot 2^{-3}$

– ez az előző szám -1-szerese, ezért annak ábrázolásától csak a 6. bájton különbözik,
a 6. bájt tartalma: 11001100_2 (= 204)

A dupla pontosságú számábrázolás az elmondottaktól csak annyiban különbözik, hogy a mantissza nem 3, hanem 7 bájton ($7 \cdot 8 = 56$ biten) *van elhelyezve.*

Itt szeretnénk felhívni a figyelmet a lap 1984. 3. számának 17. oldalán található 3. táblázatra.

Módosítsuk most az első program 2Ø-as sorát a következőképpen:

```
2Ø A = 7/3 : A# = 7/3 : PRINT "A =";
   A,"A# =";A# : PRINT
```

és futtassuk le. Az eredmény:

```
A = 2.33333      A# = 2.3333332538Ø0474
4 Ø 65 85 85 21 13Ø
8 Ø 65  Ø  Ø  Ø  Ø 85 85 21 13Ø
```

Végezzük el most a következő módosítást a 2Ø-as sorban:

```
2Ø  A = 7/3 : A# = 7#/3 : PRINT  "A =
=";A, "A# =";A# : PRINT
```

és futtassuk le így is a programot. Az eredmény:

```
A =2.33333      A# = 2.333333333333333
4 Ø 65 85 85 21 13Ø
8 Ø 65 85 85 85 85 85 85 21 13Ø
```

Az első példában a számítógép a 7/3-ot mindkét esetben egyszeres pontossággal számolta ki. Ezt az értéket az A# változóba ugyanilyen pontossággal helyezte el, a mantissza további négy bájtját nullákkal töltötte fel. Az így ábrázolt szám tehát a

$Ø.10010101\ Ø1010101\ Ø1010101\ Ø1010101_2 \cdot 2^2 =$

$$= \left(\frac{1}{2}+\frac{1}{2^4}+\frac{1}{2^6}+\frac{1}{2^8}+\ldots+\frac{1}{2^{24}}\right)\cdot 2^2 = \frac{9786709}{4194304}$$

Amikor az A és az A# változók értékét kiírattuk, akkor a számítógép a fenti közelítő tört értékét 6, illetve 16 jegy pontossággal adta meg.

A második példában a 7-es után írt számjel (#) hatására a gép az osztást dupla pontossággal végezte el, és ezt a számítási eredményt helyezte el az A# változóba. Ekkor tehát a következő számot ábrázolta:

$Ø.10010101\ Ø1010101\ Ø1010101\ Ø1010101\ Ø1010101\ Ø1010101\ Ø1010101\ Ø1010101_2 \cdot 2^2 =$

$$= \left(\frac{1}{2}+\frac{1}{2^4}+\frac{1}{2^6}+\frac{1}{2^8}+\ldots+\frac{1}{2^{56}}\right)\cdot 2^2 =$$

$$= \frac{42033596522124629}{18014398509481984}$$

Ez a tört pedig a 7/3-ot már legalább 16 jegy pontossággal közelíti, és ezt a közelítő értéket láthatjuk a képernyőn.

Felsorolunk néhány további példát, ahogy a 7/3 közelítő értékét 16 jegy pontossággal megkapjuk:

```
A# = 7#/3 : PRINT A#
PRINT 7#/3
PRINT 7DØ/3
A# = Ø.7# : A# = A#/Ø.3#
A# = Ø.7DØ : A# = A#/Ø.3#
```

Az elmondottak alapján már megmagyarázhatók a bevezetőben említett példák, és további érdekes és értékes tapasztalatok is nyerhetők.

GYAPJAS FERENC–MAJOR ZOLTÁN

HT-1080Z

GÉPI KÓDÚ KIEGÉSZÍTÉSEK BASIC-HEZ

Ismert dolog, hogy a SYSTEM/12288 bővítés meghívása után, ha BASIC program futása közben lenyomunk valamilyen billentyűt, a programfutás kis időre, kb. fél másodpercre megáll, ami például játékprogramok esetén nagyon kellemetlen.

Ez ellen többféleképpen lehet védekezni:

- Nem hívjuk meg a bővítést, vagy ha már megvan, akkor a gépet kikapcsoljuk.
- Ha mégis szükség van a kisbetűkre, SYSTEM/12294 begépelésével hívjuk a bővítést. Ekkor csak kisbetűk lesznek.
- Mivel nem tarthatjuk számon minden programunkról, hogy bővítéssel jó-e vagy anélkül, az ilyen programok elején meghívjuk azt a gépi rutint, amely a megfelelő rendszerváltozók áttöltésével „megszabadít" minket a bővítéstől.

A gépi rutin assembler nyelven a következő:

```
LD HL,06D2H
LD DE,4000H
LD BC,0036H
LDIR
RET
```

Ezt a kis rutint legegyszerűbb egy REM sorban tárolni, ugyanis a program kimentésekor így magával viszi a gépi rutint is.

A REM sor sorszáma 0 legyen, így kisebb a valószínűsége, hogy véletlenül eléírunk más sort. Ez azért fontos, mert a REM sor címének állandónak kell lennie.

A REM után írjunk be 14 db szóközt, ennek helyére fog kerülni a gépi rutinunk.

Ezután hívjuk meg a monitort SYSTEM/12710 beírásával, majd írjuk: M42EE, azaz a 42EEH címre fogunk beírni. Ez az első BASIC programsor első utasítása utáni első karakter címe. Most a következő adatokat írjuk be (NEW LINE nem kell):

```
21 D2 06 11 01 40 1D 01 36 01 05
ED B0 C9
```

X billentyűvel kilépünk az M üzemmódból, B-vel a monitorból. Írjuk be a következő programsort:

```
1 POKE 16526,238:POKE 16527,66:
A = USR(0)
```

Ezzel kész is van. Bármely BASIC program, amely így kezdődik, RUN-nal indítva kikapcsolja a bővítést.

UHERKOVICH PÉTER

Kedves Olvasónk!

Kéziratokat (amiket nem rendeltünk meg) nem őrzünk meg és nem küldünk vissza. Levelek, cikkek stb. közlése esetén szerkesztőségünk fenntartja magának a jogot az írások rövidítésére.

Két év telt el azóta, hogy az iskolaszámítógép-program keretében a középiskolákba megérkeztek az első számítógépek. Nem egyforma fogadtatásban részesültek, és nem is ugyanazt a hatást váltották ki minden iskolában. Nem tölthetik be a számítógépek a nekik szánt szerepet ott, ahol hét lakat alatt elzárva őrzik őket, és csak szigorúan meghatározott időben lehet hozzájuk férni. Annál nagyobb érdeklődést ébreszthetnek ellenben a számítástechnika iránt azokban az iskolákban, ahol tanár és diák bármikor leülhet mellé jük, ahol értő irányítás mellett ismerkedhetnek és dolgozhatnak velük a tanulók. Ilyen körülmények között a számítógép körül általában felpezsdül az élet, nagy a lelkesedés, és sorra születnek a szebbnél szebb eredmények. Mint például abban a két iskolában, amelyről a következő beszámolók szólnak.

Ahol élnek a lehetőségekkel

BUDAPESTEN

Iskolánkban, az Ihász Dániel Ipari Szakközépiskolában az 1982/83-as tanév végén jelent meg az első személyi számítógép. A következő tanévben két szakkörben ismerkedtek a tanulók a számítógéppel és a BASIC nyelvvel, és hamarosan ontani kezdték az oktatást segítő kisebb-nagyobb programokat.

A diákok lelkesedésére jellemző, hogy a szakköri foglalkozásokon a második-harmadik alkalom után a játékprogramok már elő sem kerültek. Nem az érdeklődés felkeltése, hanem az okozott gondot, hogy a szakkörösök megtalálják a helyes arányt a tanulás és a programozás között. Néhányan legszívesebben még éjjel is az iskolában maradtak volna a számítógép mellett.

A tanárok megnyerésére bemutató szakórát rendeztünk, amely a számítógép tanórai alkalmazhatóságát bizonyította. Két tanulónk nyári számítógépes táborban vett részt, ahol a gépi programozás alapjaival ismerkedtek meg.

Az 1984/85-ös tanévben az iskolai számítástechnikai munka kiszélesedett. Bekapcsolódott ugyanis egy BASIC alaptanfolyamot végzett kolléga, és az elmúlt évben legeredményesebben szereplő két harmadik évfolyamos tanuló is szakkört kezdett vezetni. Így az idén már négy szakkör működik. Bebizonyosodott, hogy a lelkesedés nem szalmaláng volt, nem az újdonság varázsának szólt.

Felvettük a kapcsolatot bázisvállalatunk, a Hungarocamion szervezési osztályával, ahol Commodore 64-es számítógépeken tartják nyilván a járműállomány futási-javítási adatait. Munkájukat programokkal kívánjuk segíteni, azzal a nem titkolt vággyal, hogy előbb-utóbb kapunk tőlük egy gépet használatra.

Új színfoltot jelentett a tanulóifjúságot megmozgató tanulmányi versenyek sorában a programpályázat, amelyet hazánk felszabadulásának 40. évfordulója tiszteletére írtunk ki.

Novemberben házi alapfokú programozási tanfolyamot indítunk iskolánk tanárai és a gazdasági osztály dolgozói számára.

Továbbra is legnagyobb problémánk, hogy a diákok és tanárok nagyon szűkre szabott időben dolgozhatnak, mert jelenleg is csak az induláskor kapott, egyetlen számítógép áll rendelkezésünkre.

MÓCZÁR LÁSZLÓ

DEBRECENBEN

A Péchy Mihály Építőipari Szakközépiskolában szerencsésen találkozott a vezetés támogatása, az új iránt fogékony tanári kezdeményezőkészség és a tanár–diák jó munkakapcsolata, amikor 1983 májusában megkapták az első számítógépet a TII központi osztásából. Itt már voltak előzmények: az iskolában korábban is foglalkoztak számítástechnikával, igaz, csak programozható zsebszámológéppel, a PTK 1050-nel. Ezek a kis gépek nemcsak szakköri keretek között működtek, hanem a matematikaórák hasznos eszközei voltak, és ma is azok.

A HT–1080Z megjelenése nem jelentett technikai csodát, természetes volt a továbblépés. Az újszülött nagy becsbe került. Külön helyiséget kapott, és tanárok, diákok időkorlátozás nélkül foglalkozhattak vele. A tanév végére több alkalommal bemutatókat szerveztek, lehetett játszani, ismerkedni a programozás rejtelmeivel. Egész nyáron nyitva volt az iskola kapuja.

A PTK-szakkörökből és az új érdeklődőkből alakult meg az első számítástechnikai szakkör. Hetente két órát foglalkoztak elméleti kérdésekkel a gyerekek, és 3–4 fős csoportokban oldották meg a szakköri feladatokat. Nyolc foglalkozás elég volt ahhoz, hogy megtanulják kezelni a gépet, egyszerű grafikai programokat írjanak.

1983 novemberében két részre oszlott a szakkör. A jobbakból haladó csoport alakult, ők ma már a gépi kódot tanulják. De a lemaradók sem vesztették el a kedvüket, ők is lelkes géphasználók.

1984 tavaszán újabb kezdő szakkört indítottak. Megvolt a technikai bázis is, az iskola két új HT-gépet kapott. Nem érdemtelenül, mert itt az átlagosnál többet futott a gép, éltek minden lehetőséggel: verseny, pályázat, KISZ játékakció, számítógépes bemutató óra fizikából. A tavaszi BNV-n is szerepeltek: önálló programjaikat mutatták be a tanulók.

A szakkörök vezetője Száldobágyi Zsigmond tanár. Az ő irányításával jutottak el oda, hogy ma már a haladó szakkör önképzőkörként dolgozik. Valamennyi programjuk, a legegyszerűbbtől a bonyolult építészeti tudást igénylőig, mind önálló munka.

A szakkör tagjai közül nyáron hatan a velencei, ketten a hajdúszoboszlói számítástechnikai táborban voltak.

Esetenként előfordult, hogy szülők, kollégák felvetették: nem terheli-e túl a gyerekeket a sok számítógépezés? Ez az aggodalom azonban alaptalan volt. Várakozáson felüli tanulmányi és érettségi eredmények igazolták, hogy az értelmes, komoly szellemi munka kedvezően hat a tanulmányi eredményekre.

Az 1984/85-ös tanév új feladatot és új lehetőséget hozott, ami nem kis örömet jelent. A TII jóvoltából – mintegy elismerésként a figyelemreméltó eredményekért – egy Commodore 64 típusú gépet kapott az iskola.

Okosan élnek a lehetőségekkel. Jól hasznosítják a külső segítséget, érkezzen az a tanügyigazgatástól vagy a Megyei Pedagógus Továbbképzési Intézettől. Készülnek a számítástechnika tanterv szerinti oktatásának bevezetésére, felmérik diákjaik majdani munkahelyeinek számítástechnikai gyakorlatát, hogy azzal is megismertessék tanulóikat.

GALGÓCZY IMRE

Kedves Olvasónk!

Szíves tájékoztatására közöljük a Lapkiadó Vállalat azon budapesti és vidéki üzleteinek címét, ahol lapunkat, a Mikroszámítógép Magazint megvásárolhatja:

BUDAPEST

VII., Almásy tér 11.
VII., Dohány u. 82.
VII., Dohány u. 84.
IX., Üllői út 119.
XII., Böszörményi út 3/a.
XIII., Rajk László u. 24.
XIV., Thököly út 21.

VIDÉK

Bükkfürdő, pavilonsor
Debrecen, Simonfi u. 1.
Győr, Árpád út 63.
Kőszeg, Jurisics tér 10.
Sopron, Orsolya tér, Lábasház
Szeged, Bajcsy-Zsilinszky u. 19.
Szombathely, Berzsenyi Dániel tér 3.
Pécs, Jókai u. 25.

FFGT

FLOPPYMAT FEJLESZTŐ GAZDASÁGI TÁRSASÁG

VÁLLALATOK, SZÖVETKEZETEK, INTÉZMÉNYEK RÉSZÉRE AJÁNLJA

– alkalmazásban levő, kipróbált mikroszámítógépes adatfeldolgozó rendszereinek széles választékát,

– újabb – a választékban nem szereplő – rendszerek kifejlesztését, bevezetését,

– tanácsadást meglévő ügyvitel korszerűsítésére, a hatékonyság növelésére

AZ ALKALMAZOTT GÉPTÍPUS
(a Floppymat gépcsalád) a kategóriájában kitűnik:
– kiemelkedő adatfeldolgozási sebességével,
– igen rugalmasan változtatható konfigurációjával,
– viszonylagos olcsóságával (célgép már 245 eFt-tól)

NÉHÁNY BEVÁLT RENDSZERÜNK

– SZÖVEGFELDOLGOZÁS
– BOLTI ELSZÁMOLTATÁS
– TERMELÉS ÜGYVITEL
– TÖBBRAKTÁRAS KÉSZLETNYILVÁNTARTÁS, -ELSZÁMOLÁS

SZOLGÁLTATÁSAINK

– ALACSONY ÁRFEKVÉS
– AZONNALI SZÁLLÍTÁS
– BEVEZETÉS, BETANÍTÁS
– SZÉLES KÖRŰ REFERENCIA
– KÍVÁNSÁGRA HELYSZÍNI BEMUTATÓK TARTÁSA

Tagvállalatok:

ECONORG
SZÖVORG
VILATI

Felvilágosítással szolgál:

Krepler Károly	127-628
Sólyom József	189-162
Szabó József	327-748

COMPUTER–M

Újabb színfolttal gazdagodott a számítástechnikai kiskereskedelem. COMPUTER–M néven megnyitotta első ügyfélszolgálati irodáját a KSH Számítástechnikai és Ügyvitelszervező Vállalat Budapesten, a VI. kerület, Lenin krt. 57–59. szám alatt. Az iroda patinás környezetben kínálja a legmodernebb technika eredményeit – a kisiskolástól kezdve a professzionális alkalmazókig, mindenkinek.

A nyitás előtt néhány nappal beszélgettünk dr. Kondricz Józseffel, a SZÜV vezérigazgatójával.

– Mit írnának képzeletbeli cégtáblájukra, ha az iroda megnyitásának motivációját kellene megfogalmazni?

Az elsődleges cél a számítástechnika-alkalmazás támogatása, fórum teremtése a számítástechnika megfelelő színvonalú terjesztésére, kiskereskedelmi hálózatunk megalapozása. Egyrészt lakossági szolgáltatás, másrészt előretolt helyőrségünk üzleti partnereink számára.

– Mit kínálnak a lakosság, a magánember részére?

Lakossági szolgáltatásaink a mikroszámítógépekhez és a tájékoztatáshoz kötődnek. Az irodában 6 mikrogépes munkahelyet alakítottunk ki, egyrészt professzionális, másrészt személyi számítógépek használatára. 2 db MO8X, 2 db TAP34 és 2 db Commodore 64 számítógéprendszer áll az érdeklődők és fejlesztők rendelkezésére. Az abszolút kezdők szakembereink segítségével ismerkedhetnek meg ezekkel a gépekkel. Akik saját fejlesztésre akarják használni, díjazás ellenében vehetik igénybe az iroda gépparkját. A professzionális mikrogépek bérleti díja egy órára 170 forint, a Commodore 64 gépé 100 forint. A diákok kedvezményt élveznek.

– Milyen szolgáltatásokat tartalmaz ez a bérleti díj?

Természetesen a szükséges alapszoftverek használatát, például Commodore gép esetében a CP/M modult is. Ezenkívül kézikönyvek, dokumentáció használatát és tanácsadást. A fejlesztő használhatja saját adathordozóját, vagy a helyszínen vásárolhat. A hajlékonylemezek ára típustól függ. A legalacsonyabb ár 10 darabos csomagonként 1200 forint körül van.

És ide tartozik egy másik jó hír is a mikrogépes fejlesztőknek. A mikronyomtatóba való leporelló is kapható: 1000 lap ára 470 forint. A leporellót csak korlátozott mennyiségben kívánjuk itt árulni, kifejezetten az egyéni fejlesztők számára.

A szakkönyvek, kézikönyvek is megvásárolhatók. Mintegy 50 féle szakkönyv kapható szervezési, programozási témakörben a Műszaki Könyvkiadó, az LSI, a Statisztikai Kiadó és a SZÁMALK könyvkínálatából. Kaphatók a számítástechnikával foglalkozó újságok, magazinok is.

– Mi az, amiért a professzionális felhasználónak vagy a számítástechnikát alkalmazni kívánó vállalatnak érdemes ide betérni?

Ingyenes szaktanácsadást adunk a megvalósítandó feladatokhoz illeszkedő gépkiválasztás kérdésében, amennyiben ehhez az üzletben történő konzultáció elegendő. Ezen túl vállaljuk a vállalat átvilágítását is, megfelelő díjazásért. Az iroda mögött áll a SZÜV több száz fős szervezőszakember-kapacitása. Irodánkban MO8X és TAP34 típusú mikrogépek vásárolhatók minta után, bevizsgált alap- és felhasználói szoftverekkel együtt. Géptípusonként 10–10 alkalmazói szoftvert kínálunk.

Nemcsak a SZÜV szellemi kapacitása, hanem nagyszámítógépes eszközbázisa is a felhasználó rendelkezésére áll új irodánkon keresztül. A helyszínen vállaljuk a Commodore 64-gyel előállított adatállományok alkalmassá tételét nagyszámítógépes rendszerekbe történő bevitelre, mágnesszalagra történő konvertálás útján. Terveink között szerepel az iroda adatátviteli vonalon való összekapcsolása a SZÜV budapesti számítóközpontjával, ahol R35, IBM és Siemens gépek fogadják a bemenő adatokat.

– Milyen szolgáltatást várhatnak az irodától a konkurrens számítástechnikai vállalatok?

Mint említettem, alapvető feladatunknak tartjuk magának a számítástechnika-alkalmazásnak fórumot teremteni. Bármely vállalat termékeit kérésre bemutatjuk, szolgáltatásait terjesztjük. Ebből a szempontból irodánk bemutatóterem is.

– Tud-e valami érdekeset mondani az iroda működéséről, terveiről?

Legfőbb erősségünknek tartjuk az iroda 7

fős személyi állományát, amely kizárólag magasan kvalifikált, nagy szakmai tapasztalattal rendelkező mérnökökből, rendszerszervező közgazdászokból és informatikai szakemberekből áll. Szakmánkhoz kapcsolódik, hogy a kezdeti tapasztalatok után az iroda teljes adminisztrációját számítógépen kívánjuk elvégezni. Erre annál is inkább szükség van, mert még az idén két újabb iroda kezdi meg működését Kecskeméten és Szombathelyen, a következő években pedig a SZÜV minden vidéki egységéhez szeretnénk kialakítani ezt a fórumot.

ZÁRDA SÁRA

Még egyszer az APL-ről

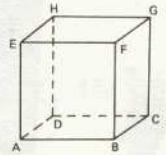
Nagy örömmel olvastam az 1984/6. számban Lovrics László MELEDA az APL-ben című cikkét. Jól mutatja be, hogy az APL mennyire áttekinthető, a részleteket eltakaró, csak a lényegre törő programokat eredményezhet, amelyek megfogalmazása és megírása nem kíván nagy szellemi koncentrációt.

Ugyanakkor fájlalom a szerzőnek azt a megállapítását, hogy „APL-lel könnyű áttekinthetetlen és józserivel módosíthatatlan programokat irni...". Ez akkor igaz, ha már a programtervezés során olvashatatlan (erről is kellene vitázni!) és pornográfikus programok írását tűztük ki célul.

Az ellenkező irányzat bemutatására közlöm a Számítástechnika című lap 1984. 1–2. számában megjelent feladatot (Hack Frigyes, ELTE):

23. számú feladvány

Az ábrán látható kocka drótból készült élhálózatán egy hangya mozog egyenletesen. Egy élen egy másodperc alatt megy végig. Az A csúcsból indul el, és az átellenes G csúcsban elhelyezett táplálékot keresi. A kocka-



1. program

```
    ∇ KOCKA N
[1]  ⍝           Ez a program a 'Szamitastechnika' 23. sz.
[2]  ⍝           feladvanyanak (1984. 1-2) megoldasa
[3]  ⍝
[4]  ⍝           Eloszor megszamozzuk a kocka sarkait az alabbi meg-
[5]  ⍝           feleltes szerint, mert igy a megoldas hajszalnyi-
[6]  ⍝           val konnyebb:
[7]  ⍝                A B C D E F G H
[8]  ⍝                1 2 3 4 6 7 8 5
[9]  ⍝           Tehat a celsarok, a 'G' kapja a legnagyobb szamot.
[10] ⍝
[11] ⍝           Most elkeszitjuk a lehetseges utak 7×3-as matrixat,
[12] ⍝           ahol az n-edik sorban talalhato szamok
[13] ⍝           azt mutatjak, hogy az n-edik sarokbol
[14] ⍝           melyik sarokba lehet eljutni:
[15]  UT← 7 3 ⍴ 2 4 6 , 1 3 7 , 2 4 8 , 1 3 5 , 4 6 8 , 1 5 7 , 2 6 8
[16] ⍝           A 'T' idok gyujtesere szolgalo TGY gyujtot
[17] ⍝           uresse tesszuk:
[18]  TGY←⍳0
[19] ⍝           IN szamolja a mar elvegzett kiserleteket:
[20]  IN←0
[21] ⍝           A program argumentuma, N, adja meg az elvegzendo
[22] ⍝           kiserletek szamat:
[23] CIKN:→(N<IN←IN+1)/NVEG
[24] ⍝           Mindig az egyes (A) sarokbol indulunk,
[25] ⍝           R fogja tartalmazni a megtett utat:
[26]  R←1
[27] ⍝           Itt vegezzuk el az uj ut generalasat, es hozzakap-
[28] ⍝           csoljuk az eddig megtett uthoz. A generalasnal
[29] ⍝           az R utolso eleme altal meghatarozott sorban
[30] ⍝           levo szamok kozul veletlenszeruen valasztunk egyet:
[31] CIKI:R←R,UT[¯1↑R;?3]
[32] ⍝           Ha az utoljara kapott sarok nem a 8-as (G),
[33] ⍝           akkor folytatjuk az 'utkeresest' :
[34]  →(8≠¯1↑R)/CIKI
[35] ⍝           Kiiratjuk a megtalalt utat:
[36]  'ABCDHEFG'[R]
[37] ⍝           A kiserlet soran kapott T ido eggyel kevesebb,
[38] ⍝           mint az utban levo csomopontok szama; T uj erteket
[39] ⍝           hozzafuzzuk TGY-hez:
[40]  TGY←TGY,¯1+⍴R
[41] ⍝           Ujabb kiserletet kezdunk el:
[42]  →CIKN
[43] ⍝           Megkeressuk a legnagyobb idot:
[44] NVEG:MAX←⌈/TGY
[45] ⍝           Meghatarozzuk a lehetseges mozgasi idok
[46] ⍝           gyakorisagat, FT-t (az osszes lehetseges idot
[47] ⍝           'A' tartalmazza):
[48]  FT←+/(A←2+⍳MAX-2)∘.=TGY
[49] ⍝           Kiiratjuk a gyakorisagi tablazatot:
[50]  A,[0.5] FT
[51] ⍝           Kiiratjuk a mozgashoz felhasznalt ido
[52] ⍝           atlagos erteket:
[53]  'ATLAG= ',⍕(+/TGY)÷N
[54] ⍝           A gyakorisagokrol - grafikon helyett tobbet mutato
[55] ⍝           hisztogrammot keszitunk, megfeleloen formazva:
[56]  (((⍴A),3)⍴ 3 0 ⍕A),' ',⍤1 '[1+FT×.<⍳⌈/FT]
[57] ⍝
[58] ⍝           A program IBM 5110 szamitogepen, APL nyelven keszult,
[59] ⍝           tervezesi,irasi es tesztelesi ido: 36 perc
[60] ⍝           Keszitette: Kertesz Adam, Bp. XI. Vahot u. 6
    ∇
```

2. program

```
    ∇ KOCKA9 N
[1]   T←⍳(I←0),0⍴U← 7 3 ⍴ 2 4 6 , 1 3 7 ,
      2 4 8 , 1 3 5 , 4 6 8 , 1 5 7 , 2 6
[2]   X:→(N<I←I+R←1)/Z
[3]   Y:→(8≠¯1↑R←R,U[¯1↑R;?3])/Y
[4]   'ABCDHEFG'[R]
[5]   →X,0⍴T←T,¯1+⍴R
[6]   Z:A,[0.5] F←+/(A←1+2×⍳⌊0.5×⌈/T)∘.=T
[7]   'ATLAG= ',⍕(+/T)÷N
[8]   (((⍴A),3)⍴ 3 0 ⍕A),' ',⍤1 '[1+F×.<⍳
    ∇
```

3. program

```
          KOCKA 30
ADHDADHEHEHG
ADADCG
AEABCG
ADCG
ABFBFEAEHG
AEHG
ADCDCBFEFBFBCG
AEADCG
AEHDHEABFBAEHEFG
ADHG
ABAEHEHDCG
ADCDAEADABFG
ADAEHEFEFBCBFBFEAEFBAEAEHEABFEHEHDHG
ABCDHDAEHDHDABCG
AEHEHDABAEHG
ADCDHG
AEFBABCBFG
ABADHDCBFBADHDABCBFG
ADCDHDAEFBFBFEHEABAEFEFG
AEADHEHDCG
AEHG
ADCDCBFEFBFBCG
ABCBCBABAEADHDAEAEABFG
ABAEAEADCDAEFEABFG
AEFBFEABADCG
ABCDHG
ADADCDABFBCDCBCBCDCG
ABCBAEFG
AEFG
AEHEFG
  3  4  5  6  7  8  9 10 11 12 13
 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35
  5  0  6  0  1  0  4  0  4  0  2
  0  2  0  1  0  2  0  1  0  1  0
  0  0  0  0  0  0  0  0  0  0  1
ATLAG= 10.733
  3 □□□□□
  4
  5 □□□□□□
  6
  7 □
  8
  9 □□□□
 10
 11 □□□□
 12
 13 □□
 14
 15 □□
 16
 17 □
 18
 19 □□
 20
 21 □
 22
 23 □
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35 □
```

```
      KOCKA9 30
 ADHDADHEHEHG
 ADADCG
 AEABCG
 ADCG
 ABFBFEAEHG
 AEHG
 ADCDCBFEFBFBCG
 AEADCG
 AEHDHEABFBAEHEFG
 ADHG
 ABAEHEHDCG
 ADCDAEADABFG
 ADAEHEFEFBCBFBFEAEFBAEAEHEABFEHEHDHG
 ABCDHDAEHDHDABCG
 AEHEHDABAEHG
 ADCDHG
 AEFBABCBFG
 ABADHDCBFBADHDABCBFG
 ADCDHDAEFBFBFEHEABAEFEFG
 AEADHEHDCG
 AEHG
 ADCDCBFEFBFBCG
 ABCBCBABAEADHDAEAEABFG
 ABAEAEADCDAEFEABFG
 AEFBFEABADCG
 ABCDHG
 ADADCDABFBCDCBCBCDCG
 ABCBAEFG
 AEFG
 AEHEFG
   3  5  7  9 11 13 15 17 19 21 23
  25 27 29 31 33 35
   5  6  1  4  4  2  2  1  2  1  1
   0  0  0  0  0  1
ATLAG= 10.733
   3 00000
   5 000000
   7 0
   9 0000
  11 0000
  13 00
  15 00
  17 0
  19 00
  21 0
  23 0
  25
  27
  29
  31
  33
  35 0
```

4. program

LOKÁLIS VÁLTOZÓK BASIC PROGRAMOKBAN

A mikrogépes BASIC nyelvek gyakran *csak globális* változónevek használatát teszik lehetővé, ráadásul ezek szintakszisa (legfeljebb két szignifikáns karakter) nem sok lehetőséget ad a változók olyan megnevezésére, ami jól emlékeztetne funkciójukra. Nagyobb programok írása és még inkább módosítása során ez könnyen arra vezethet, hogy egy-egy változót a program különböző szubrutinjaiban inkonzisztens módon használunk. Az alábbi egyszerű konvenció segítheti az ilyen hibák elkerülését.

1. A program minden egyes szubrutinjához egy-egy *szintszámot* rendelünk a következőképpen:

- Ø szintűek azok a szubrutinok, amelyek nem hívnak más szubrutinokat;
- n szintűek azok a szubrutinok, amelyek legfeljebb n − 1 szintű szubrutinokat hívnak.

Tegyük fel például, hogy egy program szubrutinszerkezetét az *ábrán* látható sémával lehet ábrázolni. Az ábra szerint SB nem hív szubrutint, így Ø szintű, SA szintje 1, az FPR főprogramé pedig 2.

Az így kialakított szintek száma egy-egy programban egy jól meghatározott N érték (valószínűleg 10-nél kisebb). A főprogram szintje a legmagasabb. (Az egyes szubrutinok szintszámait ajánlatos a szubrutint bevezető kommentárban is rögzíteni!)

2. A változónevekre vonatkozó megállapodás legyen a következő: a Ø szintű szubrutinok lokális változói kezdődjenek például I betűvel, a következő szintéi J-vel, azután K-val, ... stb.

3. A szubrutinok hívási-visszatérési paraméterei *nem kezdődhetnek* az előbbi betűkkel, hiszen éppen ez teszi lehetővé, hogy egy-egy szubrutin véletlenül se rontsa el egy másik szubrutin helyes működését.

Ez a konvenció jól alkalmazható minden olyan nyelvben, amely csak globális változóneveket támogat. Természetesen ára is van: csak akkor vezethető be, ha programunk szubrutinszerkezetét alaposan átgondoltuk, megterveztük. Ez viszont – függetlenül a változónevek megválasztásától – mindenképpen célszerű!

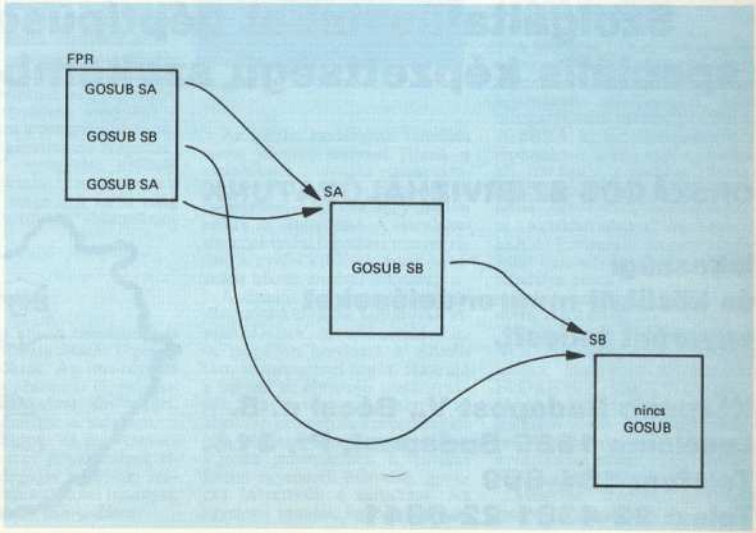
Előfordulhat olyan programmódosítás, amelynek során egy S1 szubrutinba újabb szubrutinhívást kell beiktatnunk. Ha az újabban hívott S2 szubrutin n2 szintje a hívó n1 szintjénél nem kisebb, akkor programunk szintszámrendszere megváltozna, hiszen az S1 és az őt közvetlenül vagy közvetve hívó összes szubrutin szintje megemelkedne.

A szintmódosító szubrutinhívást úgy kell végrehajtani, mintha a hívó S1 szubrutin szintje magasabb lenne a hívotténál, ugyanakkor különbözne a már meglevő magasabb szintek mindegyikétől. Nevezzük ezt a szintet az n2 + 1 = n3 szint egy *alszintjének*. Az S1 szubrutin lokális változóit az S2 hívása előtt el kell menteni az n3 szint megfelelő alszintjére. S2-ből visszatérve pedig vissza kell állítani az elmentett értékeket. Az alszinteken futó szubrutinok változóinak neveit – megint csak önkényesen – kezdhetjük például P, Q, R ... betűkkel.

Az ilyenfajta programmódosítás meglehetős körültekintést igényel, hiszen előfordulhat, hogy az S2 hív olyan n1 szintű S1' szubrutint, amelybe egyszer egy szintmódosító S2' szubrutinhívást kell beiktatnunk, így S1 és S1' elronthatják egymás elmentett lokális változóit.

Nyilvánvaló, hogy a vázolt névkonvenció nem helyettesíti a valódi lokális-globális szimbólumkezelést, de segítheti könnyebben módosítható, megbízhatóbb programok írását.

DEBRECZENI ISTVÁN



háló bármelyik csúcsából egyenlő valószínűséggel indul el az adott csúcshoz csatlakozó bármelyik élen, tehát akár visszafelé is, oda, ahonnan érkezett. Kivétel a G csúcs, melyet elérve 0 valószínűséggel megy tovább, tehát megáll. Nyilván az A csúcsból leghamarább 3 sec alatt éri el a G csúcsot. Lehetséges azonban, hogy nem a legrövidebb utat választja, sőt az esetleges visszafordulások miatt is veszít időt.

Készítsünk programot, mely

1. szimulálja a hangya bolyongását a kocka élhálózatán, és meghatározza a célbaéréshez szükséges T időt, továbbá szimulálja az A-ból G-be tartó mozgás N számú kísérletét, és meghatározza a lehetséges mozgási idők előfordulási gyakoriságát, F(T)-t, ahol T = 3, 4, ... A program inputja a kísérletek száma;

2. a képernyőre kirajzolja az F(T) gyakoriságok grafikonját, és meghatározza a mozgáshoz felhasznált idő átlagos értékét az N számú kísérlet alapján.

A feladvány részleges, és a feladat kitűzését nem mindenben kielégítő, BASIC nyelvű megoldását a Számítástechnika 1984. áprilisi számában találhatja meg az olvasó.

Bemutatom az én megoldásomat, az olvashatót *(1. program)* és az ebből származó „olvashatatlan" (?) *(2. program)* programokat, valamint ezek futásának eredményeit *(3. és 4. program)*.

A KOCKA és KOCKA9 programok futásánál a véletlenszám-generátor kezdőértékét azonosra állítottam be, ezért egyezik meg egymással a két eredmény. A KOCKA9 írásánál már figyelembe vettem azt is, hogy a megvalósult lépésszám csak páratlan lehet.

KERTÉSZ ÁDÁM

A HÓDÍTÓ ELEKTRONIKÁBAN AZ ORSZÁGBAN MINDENÜTT AZ ÖN PARTNERE VAGYUNK!

INFORMÁCIÓTECHNIKAI VÁLLALAT

VÁLLALJUK:

ELEKTRONIKUS ÍRÓGÉPEK,
SZÁMOLÓGÉPEK,
MÁSOLÓGÉPEK, – üzembe helyezését,
SZEMÉLYI SZÁMÍTÓGÉPEK, – szerviz kiszolgálását,
KISSZÁMÍTÓGÉPEK – javítását,
ÉS SZÁMÍTÓGÉPRENDSZEREK – műszaki ellátását.

Szolgáltatásainkat géptípusokra orientált speciális képzettségű szakembereink látják el.

ORSZÁGOS SZERVIZHÁLÓZATUNK

lakossági
és közületi megrendeléseket
egyaránt teljesít.

Központ: Budapest V., Bécsi u. 8.
Levélcím: 1369 Budapest, Pf. 314.
Telefon: 184-899
Telex: 22-4381 22-6841

LOGO – a programozás-tanulás új útja

Közhely, hogy ma mindenkit érdekel a számítástechnika. Nagy felelősség nehezedik arra a mozgalomra, amely az érdeklődés hullámait igyekszik megfelelő mederbe terelni. A lényeg a *megfelelő* szón van. Lehet regiszterekkel és processzorokkal bódítani a naiv szimpatizánsokat, csak az a kérdés, meddig maradnak meg hitüknél. Lehet BASIC-et és COBOL-t oktatni a tömegnek, ha nem zavarja álmunkat, hogy holnap százezrek fognak hátat fordítani a számítógép-bálványnak.

Az embereket nem fogja izgatni a csoda, ha kiderül, hogy nem sok közük van hozzá, nem tudnak hozzá közel kerülni. Csak az a gép lehet igazán érdekes, amelyet aktívan lehet megismerni, alkotva lehet használni. Legyen figyelmeztető jel számunkra az idegen nyelvek oktatásának gyakran emlegetett krízise.

A programozás oktatásában csak olyan módszert lenne szabad tömegesíteni, amely nagy valószínűséggel sikerre vezet. Lehet, hogy jogosan büszkélkedünk az iskolaszámítógép létezésével és terjesztésével, a személyi számítógépek szabadabb beáramlásával az országba. Ideje lenne azonban komolyan elgondolkodni azon, hogyan kell megtanítani tömegméretben ezeknek az eszközöknek a használatát, valahogy úgy, ahogy a golyóstollat tartó kezet megtanítjuk a betű formálására, a szavak leírására, a gondolatok rögzítésére. E feladat megoldásához kívánunk néhány gondolatot felvetni a LOGO nyelv kapcsán.

A programozásoktatás nehézségei

A számítógép világa a tanuló programozó számára marsbéli táj, melynek minden elemét s a köztük lévő összefüggéseket önmagukban kell elfogadni, anélkül, hogy emlékeztetnének az ismert „földi" dolgokra.

Egy programozási nyelv megtanulása alapvetően azért nehéz, mert olyan elvont fogalmak és kapcsolatok megértését igényli, amelyek nehezen asszociálhatók valós fogalmakkal vagy a tudatba már beépült elvont fogalmakkal.

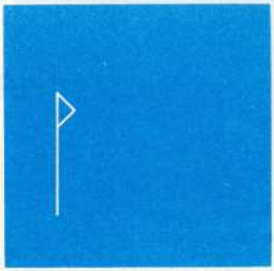
A tanulási folyamat egyik nélkülözhetetlen eleme, az *asszociatív visszacsatolás* csak elvétve kapcsolódik be a programozási nyelv elsajátításába. Kezdetben leginkább a matematika fogalmaihoz kapcsolódhatnak a programozás fogalmai, gondoljunk például az aritmetikai utasításokra. Már sokkal áttételesebben asszociálható egy tárolási művelet, beírás az operatív tár valamely rekeszébe. Még kevésbé hasonlít valami ismertre a bitmanipuláció, az eljáráshívás, a megszakítás.

A ciklust például önmagában még értené a tanuló, de FORTRAN-ban leírva és a programszövegbe ágyazva nehezen tudja felfogni értelmét. Önmagában egy utasítás legtöbbször érthető, de a programszöveg sorban következő vagy elágazó utasításai közötti kapcsolat felismerése igényli az algoritmus logikájának, szigorú következetességének ismeretét is.

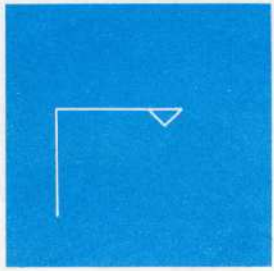
A program algoritmusa egyébként csak feltételesen kapcsolható annak a feladatnak az elvi megoldási algoritmusához, amelynek a kiszolgálására a program hivatott. Többszörös absztrakció keletkezését lehet itt megfigyelni, különösen, ha figyelembe vesszük, hogy a feladat már maga is a valós világ valamilyen vetületét, absztrakcióját jelenti.

A vizuális visszacsatolás

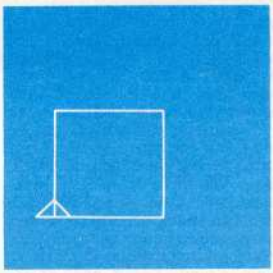
Az iskolai anyag elsajátítása is igényli az elvonatkoztatás képességét a tanulóktól. Az írás-olvasás megtanulása absztrakt elemek sorának elsajátításával jár együtt. Hasonló a helyzet a számtan, a mértan, a földrajz és más tantárgyak esetében is. A nehézségek ellenére a pedagógia mindenki számára biztosítja az iskolai tananyag megfelelő szintű befogadását.



1. ábra



2. ábra



3. ábra

Az iskolai tantárgyak tanulása során jelentős szerepet játszik a szemléltetés, a vizuális visszacsatolás, amely asszociatív folyamatok megindulását biztosítja, ily módon erősíti az elsajátítást. A szemléltetés az oktatási folyamat szerves része, a nyelvi közlések erejét a vizuális közlés erejével fokozza.

A vizuális visszacsatolás szerepe változó a különféle tantárgyak elsajátításában. Az írás-olvasás eleve magában hordozza a vizualitást, természeténél fogva. Hasonló a helyzet az ábrázoló tantárgyaknál, mint a rajz, a geometria, az ábrázoló geometria, a műszaki rajz stb. Magasabb szinten, például a legtöbb művészetben a vizuális forma egyenesen közvetíti, gyakran helyettesíti a tartalmat. Az egyetemi tanulók legtöbbje a matematika tételek bizonyításának megtanulását papíron végzi, hogy a lépéseket, a bizonyítás algoritmusát könnyebben rögzítse agyában.

A programozás elsajátítása legjobban az írás-olvasás vagy egy idegen nyelv megtanulásához hasonlítható, azzal a különbséggel, hogy az utóbbiak elsajátításának már korai szakaszában a valóság már átélt és megismert fogalmainak újraélése, asszociálása folyik.

Bár nincsenek erre vonatkozóan adataink, egyáltalán nem vagyunk biztosak abban, hogy a FORTRAN vagy a BASIC programozási nyelvek megtanítását sikerülne megvalósítani az általános iskolákban. Ennek legfőbb okát abban látjuk, hogy az említettekhez hasonló, általánosan elterjedt magas szintű nyelvek tanulása során nem keletkeznek vagy csak nagyon áttételesen olyan asszociációk, amelyek támpontot nyújtanak az absztrakt anyag elsajátításában.

Ebben a tekintetben változással kecsegtet a LOGO nyelv. (A nyelv ismertetését lásd a µM 1984/6. és 1985/1. számaiban.) A LOGO az első olyan párbeszédes programozási nyelv, amely biztosítja a nyelvi dialógus hatékony grafikai dialógussá fejlesztését egy egyszerű grafikai eszköz segítségével. A LOGO didaktikus hatása abban rejlik, hogy a programozás elsajátításának már kezdeti szakaszában megadja a szemléltetés lehetőségét, és ezzel a tanulás általában kritikus első fázisában felfedezéshez, sikerélményhez, intellektuális örömhöz juttatja a tanulót.

A LOGO vizualitása

A grafikus LOGO bekapcsolásakor megjelenik a háromszögletű rajzeszköz (teknősbéka), amelyet közérthető utasításokkal lehet mozgatni, mint például ELŐRE n, JOBBRA k. Értelemszerűen n a lépésszámot (a képernyő szélessége kb. 200 lépés), k pedig az elfordulás szögét jelenti. Ugyanígy lehet hátra, illetve balra haladva rajzolni. (Az okfejtésben az angol nyelvű LOGO kifejezések magyar megfelelőit használjuk, mivel ennek elvi akadálya nincs.)

Tehát kezdjünk el LOGO-t tanulni, 6-tól 60-ig!

Húzzunk egy 80 lépésnyi vonalat a képernyőn, aztán forduljunk jobbra. Begépeljük az utasítást: ELŐRE 80 JOBBRA 90 (lásd az *1. ábrát*). A képernyőn azonnal megjelenik a vonal. Mi történt? Begépeltünk két közérthető parancsot, s vizuálisan máris meggyőződhetünk a végrehajtásukról.

Menjünk tovább, ismételjük meg a két utasítást:

ELŐRE 80 JOBBRA 90 *(2. ábra)*. A képernyő válasza nem késik. Ismételjük ezt meg még kétszer:
ELŐRE 80 JOBBRA 90
ELŐRE 80 JOBBRA 90
Rajzoltunk egy négyzetet *(3. ábra)*.

A programozót öröm tölti el, olyan ez, mint mikor a gyereknek először sikerül az ajándék kerékpárral megtenni az utat a sarokig, és visszatérni aggódó szülőjéhez.

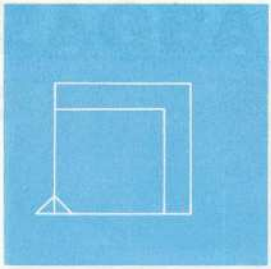
Az elvont eszközökkel leírt eljárásnak azonnal szemmel látható eredménye született.

Most összegezzük az eljárásunkat:
ELŐRE 80 JOBBRA 90
ELŐRE 80 JOBBRA 90
ELŐRE 80 JOBBRA 90
ELŐRE 80 JOBBRA 90

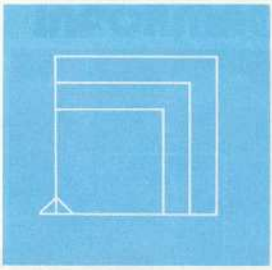
Máris motoszkálni kezd a gondolat, és az első osztályos tanulók között is sok akadna, aki megkérdezné: miért kell ugyanazt négyszer leírni, nem lehetne egyszerűbben? Az összeadás és a szorzás közötti összefüggés a leírt eljárás és a képernyőn levő ábrát tekintve nyilvánvalóan megmutatkozik. A tanító a kérdésre megkönnyebbülve feleli: lehet, persze hogy lehet egyszerűbben. Lehet ismételni, íme:
ISMÉTELD 4 ELŐRE 100 JOBBRA 90

Lerajzoltunk egy 100 lépés oldalú négyzetet *(4. ábra)*. A kisiskolások most megilletődve hallgatnak, a tanítón a meglepetés sora. „Lehet ám négyzetet rajzolni még egyszerűbben!" – szól, s legépeli: NÉGYZET, mire a képernyőn megjelenik egy harmadik, ezúttal 120 lépés oldalú négyzet *(5. ábra)*. „Hát a LOGO tudja, mi az hogy négyzet?" – csodálkoznak a kisdiákok. „Nem tudta," – válaszolja a tanító, „de én előzőleg megtanítottam neki. Mégpedig így:"
TUDD NÉGYZET
ISMÉTELD 4 ELŐRE 120 JOBBRA 90 KÉSZ

„De hát ezzel a LOGO csak 120 lépés oldalú négyzetet tud rajzolni" – veszi észre az egyik kisdiák. S így észrevétlenül eljutottunk nemcsak az eljáráshoz, hanem annak paraméterezéséhez is. Az első



4. ábra



5. ábra

órán! A tanítónak nincs is más dolga, mint bevezetni a változót (hála a korszerű oktatásnak, ez ma már a 2. osztályban természetes), és elkezdhetik a bonyolultabb ábrák rajzolását, a különböző ötletek képernyőre varázsolását.

A BASIC nyelv esetében ugyanezt a fejlődést hosszabb munkával talán elértük volna, de nem biztos, hogy mindenkinél. Jelentős mozzanatnak tartjuk, hogy a LOGO esetében nemcsak az üzemmód párbeszédes, hanem a szemléltetés révén olyan kapcsolat jön létre tanuló és tanító között, hogy az oktatási folyamatban is szinte magától kialakul a párbeszéd. (Folytatása következik.)

DR. HROTKÓ GÁBOR

VÁLLALKOZÓT KERESÜNK

1500 fős iparvállalat mikroszámítógépes bérelszámolási rendszerének kidolgozására, vagy már máshol bevezetett, hasonló rendszer adaptálására.

Az ajánlatokat az alkalmazásra javasolt számítógép-konfiguráció megjelölésével, „Pontos ügyvitel" jeligére kérjük a szerkesztőségbe.

Csébfalvi Károly, a Magyar Villamos Művek Tröszt Szervezési és Számítástechnika-alkalmazási Főosztályának vezetője a hazai számítástechnika évtizedek óta elismert tekintélye. A ma 55 éves szakember először a NIM-től kapott megbízást 1958-ban egy korszerű számítóközpont létesítésére. Ezt követően több hasonló központ (Csepel, VEIKI, MTA, KSH stb.) felállításában vett részt munkatársaival. Gyakorlati munkája mellett tevékenyen dolgozott a szakember-utánpótlás oktatásában: tanított egyetemen, mérnöktovábbképzőben, ipari továbbképző központokban.

– Az Ön egyik tanulmányában találkoztam egy különös kifejezéssel: mikro-számítástechnika. Azt kell értenünk ezen, hogy a számítástechnika bizonyos szempontból kettéágazott „nagyra" és „kicsire"?

– Nem erről van szó. A mikro-számítástechnika kifejezésre éppúgy átmenetileg lesz csak szükség, mint ahogy annak idején a mikrobarázdás lemezről beszéltünk. Hogy mégis ezt használjuk, azokra az új lehetőségekre utalunk, amelyek kihozták a számítástechnikát a géptermek elefántcsonttornyából az alkalmazások frontvonalára. A kis és nagy gépek között egyébként érdekes kettősség áll fenn: az egyre intelligensebb mikroszámítógépek egyre nagyobb, óriási adattömeget mozgató számítógépek létét is feltételezik.

– Ebben, az előbb említett átmeneti értelmezésben tapasztalatai alapján miként ítéli meg a hazai számítástechnika helyzetét?

– Abból kell kiindulnunk, hogy néhány éven belül nem lesz eladható a világpiacon olyan termék, amely valamilyen kapcsolatban ne állna a mikro-számítástechnikával. Nem csupán a gépekre, berendezésekre gondolok, hanem még a mezőgazdasági termékekre is. Enélkül ma már nincs hatékony minőségbiztosítás.

– A mezőgazdaságról szólva sok minden eszébe jut az embernek, a számítástechnika talán a legkevésbé. Hogy lehet itt felhasználni a számítógépeket?

– Például egy talajmintából meghatározható, hogy adott területen milyen növényi kultúrák termeszthetők a leggazdaságosabban és a legjobb minőségben, ehhez milyen műtrágya kell stb. Egyébként nálunk már működnek ilyen eszközökkel felszerelt laboratóriumok a mezőgazdaság megsegítésére. Ezeket kellene népszerűsíteni a „süllyeszd el a hajómat" típusú játékok helyett.

Visszatérve kiinduló tézisünkre, amelyet a piacokért folytatott, kiélezett verseny nap mint nap meggyőzően bizonyít, sajnos meg kell állapítanunk, hogy a mikro-számítástechnika hazai felhasználásában még saját lehetőségeinkhez képest is elmaradtunk.

– Állandóan új és új számítógépek üzembe állításáról hallunk. Ha ezeket nem használjuk ki megfelelően, miben látja ennek okait?

– A számítástechnika fejlődése három tényező: az emberi erőforrások, a technikai lehetőségek és a társadalmi igények egymásrahatásának eredménye. Ha külön-külön nézzük őket, elmaradásunkat döntő mértékben egyik sem okozhatja. Arra kell tehát gondolnunk, hogy a probléma a tényezők összhangjának hiányában keresendő, összefüggésben belső, strukturális aránytalanságokkal.

– Véleményem szerint egyik tényező fejlődése sem függetleníthető a másik kettőtől. Például egy magasabb színvonalú géppark beállítása elképzelhetetlen, ha erre nincs társadalmi igény, működtetése pedig lehetetlen képzett szakemberek nélkül. Kérem, fejtse ki részletesebben, milyen aránytalanságok érvényesülnek a hazai számítástechnikában?

– A szakember-hátteret vizsgálva nincs különösebb okunk panaszra. Ezt bizonyítják sikeres hazai alkalmazások, sőt külföldön dolgozó szakembereink elismerése is. Viszonylag erős a hazai elméleti kutatási háttér is. Mindezzel együtt elmaradásunk egyik okát abban látom, hogy szakembereink egy részéből hiányzik az a fajta profizmus, amely a látványos fő elemek mellett legalább akkora figyelmet fordít a kevésbé „hálás", ám használhatóság szem-

„A profizmust hiányolom...”

BESZÉLGETÉS CSÉBFALVI KÁROLLYAL

pontjából nélkülözhetetlen részletekre. Az alkalmazásnál pedig ez a profizmus azt jelenti, hogy a számítástechnikusnak szem előtt kell tartania az adott terület *valamennyi, speciális* követelményét, elsősorban az alkalmazási biztonságot. Komoly hiány van nálunk tapasztalt alkalmazástechnikai szakemberekben!

– A műszaki háttérről szólva sokan elégtelennek tartják az anyagi forrásokat. Ez volna talán az elmaradás egyik fő oka?

– A közismerten szűkös lehetőségek helyett a műszaki háttérrel kapcsolatban olyan gondokról szeretnék beszélni, amelyek gátolják meglévő lehetőségeink kiaknázását. Nincs megbízható minőségellenőrzés, amely pedig minden egyéb terméknél – a száraztésztától a vasalóig – természetes; megoldatlan a szervizhálózat, a szabványügyi tevékenység, a szoftverellátás megszervezése és még sorolhatnám a „szűk keresztmetszeteket”. Kirívó gond, hogy a gépek számának gyors növekedését kísérő euforiában elfeledkezünk a különböző járulékos alkatrészekről, csatlakozó elemekről. Minél meszszebb távolodunk a gép központi egységétől, annál szorongatóbb ez a hiányosság.

– A közhangulatból ítélve úgy tűnik, a társadalmi igény, illetve befogadókészség okozhatja a legkisebb fejfájást.

– Igen, az igények nagyobbak, mint amennyit jelenleg ki tudunk elégíteni. Ami a befogadókészséget illeti, azért nem olyan felhőtlen a kép. Sokszor lehet találkozni olyan esettel, hogy ragyogó gépet hozzá méltatlan környezetben állítanak fel. A számítógépek alkalmazását ugyanúgy tervezni kell, mint egy ház építését. Tragikomikus az „adott az Úr bárányát, majd ad hozzá legelőt is” típusú fejlesztés. Ez megint a profizmus kérdését veti fel.

– Mindannak alapján, amit elmondott, még a számítástechnikában rejlő elsődleges lehetőségeket – termelékenység-, minőségjavulás, gyorsaság stb. – sem tudtuk eddig kellő mértékben kihasználni. Hogyan tudunk akkor másodlagos, sőt további hatásokat figyelembe venni, amelyek egyes fejlettebb országokban már nem kis mértékben jelentkeznek?

– Mégis, ezekre már most fel kell készülnünk. A mikroszámítástechnika terjedése tovagyűrűző átalakulásokkal jár. Ez érinti az alapanyag-ellátást, a szerszámgépipart és más iparágakat. A munkás nélküli üzemek, a székhely nélküli intézmények lehetősége etológiai, szociológiai, munkajogi kérdéseket vet fel, de megváltozik a közlekedés, a távközlés szerepe is, hogy csak néhány példát említsek. Mindez előbb-utóbb nálunk is óhatatlanul jelentkezik, azért kell már most foglalkozni velük, hogy a strukturális változások ne érjenek készületlenül.

A jelenlegi helyzetre visszatérve: ha nem tárjuk fel ennek jellemzőit, nincs esélyünk a gyorsabb felzárkózásra. Látni kell továbbá, hogy a tárgyilagos elemzés egyben kijelöli rövid- és hosszabb távú tennivalóinkat. Egy olyan központi stratégiának megfelelően, amelynek fő elve, hogy a mikro-számítástechnika először lehetőleg a közvetlen értéket előállító anyagi termelésben jelenjen meg. Ezt a gyors, mérhető „profi” eredményre törést kell a hazai számítástechnikusainknak az eddiginél lényegesen jobban, a szemléletben és a mindennapi gyakorlatban egyaránt magukévá tenniük. Az anyagi és erkölcsi „hozam”-on túl enélkül a felzárkózás és egyáltalán a továbblépés nehezen képzelhető el.

– Szakmai körökben azt hallottam Önről, hogy annak idején sok mindent előre meglátott. Tudna-e most is mondani valamit a számítástechnikával foglalkozók számára, amit jövőbeni munkájukhoz megfontolásra ajánl?

– Nem nehéz olyat jósolni, ami „a levegőben lóg”, csak észre kell venni. Az új technika birtokbavételének lázában elterelődött a figyelem a matematika olyan új eredményeiről, amelyek éppen ehhez az új technikához tartoznak. Ma már egy sereg olyan alkalmazási területet lehet említeni, ahol ezek a jól átgondolt modellezési munkán alapuló, új matematikai módszerek vezetnek eredményre. Így például az ún. prediktor-korrektor módszerek általánosításai még a legegyszerűbb nyilvántartási, elszámolási rendszereknél is alkalmazhatók. Ha ugyanis a feldolgozás soron következő lépcsőjét egy megfelelően fejlett modell segítségével előre megmondjuk („prediktáljuk”), akkor az időközben beérkező tényadatokkal csupán a jóslást kell korrigálni.

Az elkövetkezendő években a matematika numerikus és nem numerikus módszereit rendre kézbe kell venni, és az aktuális igényeknek, valamint technikának megfelelően értékelni.

LACZKA MIKLÓS

Alapozás X.

Pneumatikus operátorunk matematikai modelljeivel foglalkoztunk a legutóbbi alkalommal. Ennek révén belekóstoltunk egy kissé magába a matematikai modellkészítésbe is. Hasznos és tanulságos lesz, ha folytatjuk e tevékenységet, még pontosabb, még egzaktabb matematikai modellek megalkotása érdekében. Emellett természetesen folytatjuk a konkrét munkát is az ún. „logikai hálózatok" építésével. Most ez utóbbi téren lépünk előbbre egyet.

Mik lehetnek a céljaink?

A digitális berendezések néhány fajta operátorból épülnek fel. Áll ez mind az elektronikus, mind a hidraulikus, mind a pneumatikus, mind pedig minden más (például fénnyel és más rezgésekkel dolgozó) digitális automatára is. Elméletileg érdekes, a gyakorlat számára pedig hasznos, és ezért kívánatos, hogy egyfajta operátorból minél többfajta egység legyen felépíthető. Az is kívánatos természetesen, hogy ezek, az így felépített operátorok ráadásul még külön-külön is minél többféle feladat megoldására legyenek alkalmasak.

Pneumatikus operátorunk *(1. ábra)* az előbbi szempontból különlegesnek bizonyul. Látni fogjuk majd, hogy – bizonyos értelemben – mindenféle feladatra való számológép-építő egység (összetett operátor) felépíthető belőle.

Fogjunk hozzá tehát bátran, és alkossunk ún. „kapcsolásokat"! Ez azt jelenti, hogy kössük össze az operátorokat a levegő zavartalan áramlását lehetővé tevő csövekkel. Az így kialakuló kapcsolásokat aztán vizsgáljuk meg, hogy hogyan működnek, mire tudjuk azokat használni.

Olyasféle ez a konstrukciós munka, mintha operátoraink atomok (ionok) lennének, és belőlük ionokat (illetve molekulákat) építenénk fel. A vegytani szerkezeti képletben ún. „karok" jelölik a közvetlen kapcsolatokat, esetünkben pedig légvezetékek. A vegytanban sokféle atom sokféle kapcsolása lehetséges, itt egyféle „atom" sokféle kapcsolása. „Univerzális" „atomunk" az 1. ábrán látható pneumatikus operátor. A kapcsolási műveletek révén létrejövő rendszert „hálózatnak" is és „kapcsolásnak" is szokták nevezni. A *logikai hálózat* és *logikai kapcsolás* szakkifejezést az indokolja – vagy csak indokolta (?!) –, hogy valamikor a logika kétértékű volt (mert csak kétértékű logikát tudtak elképzelni), és e hálózatokban is kétértékűeknek tekintünk – kezdetleges modellek alapján – egyes állapotjellemzőket.

Mielőtt szabadjára engednénk képzelőtehetségünket, és mielőtt e nagyon hasznos és szórakoztató kísérletezést elkezdenénk, célszerű azzal is foglalkoznunk, hogy egyáltalán milyen eredményre juthatunk. Milyen eredmények lennének például az ideálisak számunkra? Hátha kielégíthetők lesznek ezek az igények.

Mire volna szükségünk tehát, hogy ha például legkényelmesebben kívánnánk digitális berendezést, például számológépet építeni. E gyűjtést – nagyon hasznos munka – most az Olvasóra hagyjuk. Birkózzon vele egy kicsit! Itt csupán egyféle kívánságot említünk. Nyilvánvalóan jó lenne, ha volna például négyzetreemelést végző, összeadást végző, szorzást végző operátor. Tegyük fel, hogy e kívánság kielégíthető, és könnyen elő tudunk állítani ilyen operátorokat. Hogyan folytatnánk ebben az esetben a tervezést? A válaszadáshoz nincs szükség arra, hogy meg tudjuk mondani, hogy milyen ezeknek az operátoroknak a belseje. (Erre a kérdésre most, ebben a pillanatban nem is tudnánk felelni, később azonban ki fog derülni, hogy ez nem nehéz dolog.) Ha az operátor belsejéről nem is kell mindent tudnunk, környezetéhez való kapcsolódási lehetőségeit azonban mindenképpen ismernünk kell.

Ha tervezni akarunk, ábrázolnunk is kell a szóban forgó operátorokat. Az operátorokat nyilván most is egyetlen téglalappal jelöljük. Az is nyilvánvaló, hogy az operátorokat összekötő vezetékeket is ábrázolni kell valahogyan. Ha ennek a hagyományos, vonalas módját választjuk, akkor azt kell megmondanunk, hogy milyen lenne például egy összegképző és közvetlen környezetének vázlata. Erre a kérdésre az összegképzőre csatlakozó vezetékek megrajzolásával kielégítő választ tudunk adni. Vessünk egy pillantást a 2. ábra vázlataira! Vajon melyik vázlat lehet jó? Talán első pillanatban meglepően hangzik, hogy mindegyik vázlat jó lehet.

A vázlatok annak a módjáról adnak eligazítást, hogy hogyan történhet az operátor információs és milyen módon információhatja az operátor a környezetét. Más szavakkal ez a következőt jelenti. Hogyan bocsáthatjuk az operátor rendelkezésére az operandusokat? És hogyan bocsáthatja az operátor rendelkezésünkre az operátumot, azaz az operáció eredményét? A válasz az, hogy *sokféleképpen*. Oly sokféleképpen, hogy nincs is helyünk részletes elemzésükre. Lássunk azonban néhány kiragadott esetet, a 2. ábra elemeinek sorrendjében! Szerencsére ezeknek a szemelvényeknek alapján is könnyű elképzelni a lehetőségek gazdag sokaságát, egyúttal teljes áttekintésükre is megadják a lehetőséget. (Az ábrán x az egyik, y a másik összeadandó, z pedig az eredmény, *számjegysorozat* formájában.)

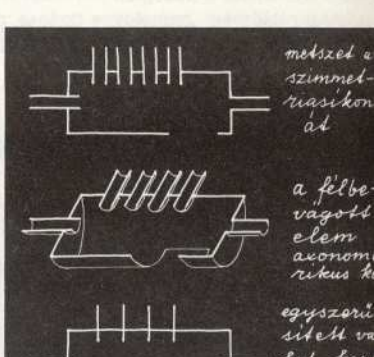
2.(1) ábra. x és y számjegysorozatának minden eleme külön-külön vezetékeken egyidejűleg áll az összegképző rendelkezésére (egy darabig). Ha az operátor elkészült az eredmény-számjegysorozattal, akkor annak minden eleme egyidejűleg rendelkezésre áll az előzőktől különböző, külön-külön vezetékeken (egy ideig).

Egy másik lehetséges magyarázat ugyanerre az esetre a következő. x és y számjegysorozatának minden eleme külön-külön vezetékeken áll az összegképző rendelkezésére, egy-egy darabig, valamikor, nem szükségképp egymással egyidőben. Az eredmény számjegysorozatának elemei az előzőktől különböző, külön-külön vezetékeken észlelhetők egy ideig, valamikor, nem szükségképp egymással egyidőben.

2.(2) ábra. x és y számjegysorozatának minden eleme külön-külön vezetékeken, egyidejűleg áll az összegképzőnek rendelkezésére (egy darabig). Az eredmény számjegysorozatának elemei egy közös, az előzőktől különböző vezetéken észlelhetők, valamikor, valamennyi ideig, egymásutáni sorrendben.

Egy másik lehetséges magyarázat ugyanerre az esetre a következő. x és y számjegysorozatának minden eleme külön-külön vezetéken áll az összegképző rendelkezésére, egy darabig, valamikor, lehet, hogy számjegyenként más-más időben. Az eredmény számjegysorozatának elemei egy közös, az előzőktől különböző vezetéken észlelhetők valamikor, valamennyi ideig, egymásután, valamilyen, nem szükségképp a sorozatban levő sorrendjükben.

E példákból már megállapíthatjuk, hogy az egyes



1. ábra

információk megjelenési (tartózkodási) helyére és megjelenési (ott-tartózkodási) idejére vonatkozó információkra különösen kell figyelnünk. A 2. ábrán szereplő példák esetében 2 információnak például észlelése, feldolgozása lehet

a) ugyanabban az időben, ugyanazon a helyen történő;

b) ugyanabban az időben, különböző helyen történő;

c) különböző időben, ugyanazon a helyen történő;

d) különböző időben, különböző helyen történő.

Esetünkben a) nem lehetséges, mivel nem lehet egy csőben (pneumatikus vezetékben) egy ponton ugyanakkor áramlás is és nem áramlás is, tehát a két különböző információ megkülönböztetésére nem volna alkalmunk. De lehetséges például egy információsorozat elemeinek időben sorban, egymás után való bejuttatása valahová úgy, hogy mindegyik elemet külön-külön, egymással (geometriailag) párhuzamos vezetékeken továbbítunk.

Ha az információk megragadása (megnevezése) csak két jellemzővel (idő, hely) történik, az a) esetben nyilvánvalóan nem valósítható meg 2 információ megkülönböztetése. Egyébként azonban nem lehetetlen a megkülönböztetés, mert például a fényinformációnak (fényimpulzusnak) hullámhossza is van, és ez alkalmas a hely és idő információ azonossága esetében 2 információ egymástól való megkülönböztetésére. Ez utóbbi esetben természetesen az azonos helyű, azonos időben jelentkező, azonos hullámhosszú információk megkülönböztetése okoz majd problémát, ami egy újabb megkülönböztetésre alkalmas jellemző szerepeltetésével már megoldható lesz.

Magyarázatra szorul még a 2. ábra (7), a (9) és a (10) esete.

Lássuk például a (10)-et! x és y számjegysorozatát képviselő információk csak egymás után jelenhetnek meg az egyetlen bemeneti vezetéken. Sorrendjük lehet például $x_1, y_1, x_2, y_2, \ldots$ vagy $x_1, x_2, \ldots, y_1, y_2, \ldots$, vagy bármi más sorrend. Az operátornak kell tudnia, hogy mikor mit érzékel.

A (11), (12), (13), (14) és (15) esetek azért érdekesek, mert kiderül belőlük, hogy a gyakorlatban nem mindig lehet egy vezetéket bemeneti vagy kimeneti vezetéknek nevezni. Ezekben a példákban egyes ve-

zetékeken egyszer bemeneti, máskor pedig kimeneti információ van (áramlik). A (15) példa esetében ezek egymás után, például az $x_1, x_2, \ldots, y_1, y_2, \ldots, z_1, z_2, \ldots$ sorrendben észlelhetők ugyanazon a ponton (ugyanabban a vezetékben).

Hangsúlyozzuk, hogy elméleti célokból meg lehet csinálni és sokszor célszerű is megcsinálni a különböző funkciójú és különböző irányú információáramlások külön-külön vezetéken való lebonyolítását, például matematikai modellekben.

Pneumatikus vezeték esetében természetesen nem lehet a vezeték egy pontja egyidőben különböző irányú információk továbbítására igénybe véve. (Fényinformációkra ez természetesen nem áll.)

Elektronikában újabban egyre gyakoribb egy-egy pont (vezeték) kétirányú használata (általában nem egyidőben). Ilyenkor ugyanazon a ponton egyszer bemeneti, máskor pedig kimeneti információk érzékelhetők.

Ha egy vonal (vezeték) kétirányú, akkor nincs értelme bemenetinek vagy kimenetinek nevezni, helyesebb a *be- kimeneti vezeték* vagy csak a *csatolóvonal, csatolóvezeték* név használata.

A 2. ábra (16) példáját csak tévedések elkerülése céljából szerepeltettük, ez az (1) példa „rövidített" változata, a vezetékek ismételt megrajzolása helyett azok multiplicitását (hányszorosságát) jelöljük a ferde áthúzás mellett.

A gyakorlatban nem ritka egyes be- vagy kimeneti pontok összekapcsolása ugyanannak az operátornak valamelyik másik be- vagy kimeneti pontjával. Ez gyakran jól használható megoldásokat szolgáltat. Elnagyolt matematikai modellek használatánál azonban gyakran előfordul az implicit kapcsolat az ilyen hálózatok matematikai modelljében. Ebből pedig ritkán szokott jó származni. Az implicit függvények egzakt matematikai kezelése, modellezésben való helyes alkalmazhatóságuk ismerete terén ugyanis nincs különösebb okunk a dicsekvésre.

Megjegyezzük, hogy a vegyes (be-ki) pontokat olyankor, amikor bemeneti pontként működnek bemeneteknek, olyankor, amikor kimeneti pontként működnek, kimeneteknek, amikor pedig „lezártak" (nem működtethetők sem bemenetként sem kimenetként), nem létezőknek tekinthetjük.

Így, összefoglalva az eddigieket, jegyezzük meg, hogy a bemeneti információknak a bemeneteken való rendelkezésre állásának (rendelkezésre bocsátásának), a bemeneti információ feldolgozásának, és a kimeneti információknak a kimeneteken való rendelkezésre bocsátásának a módja *sokféle lehet*.

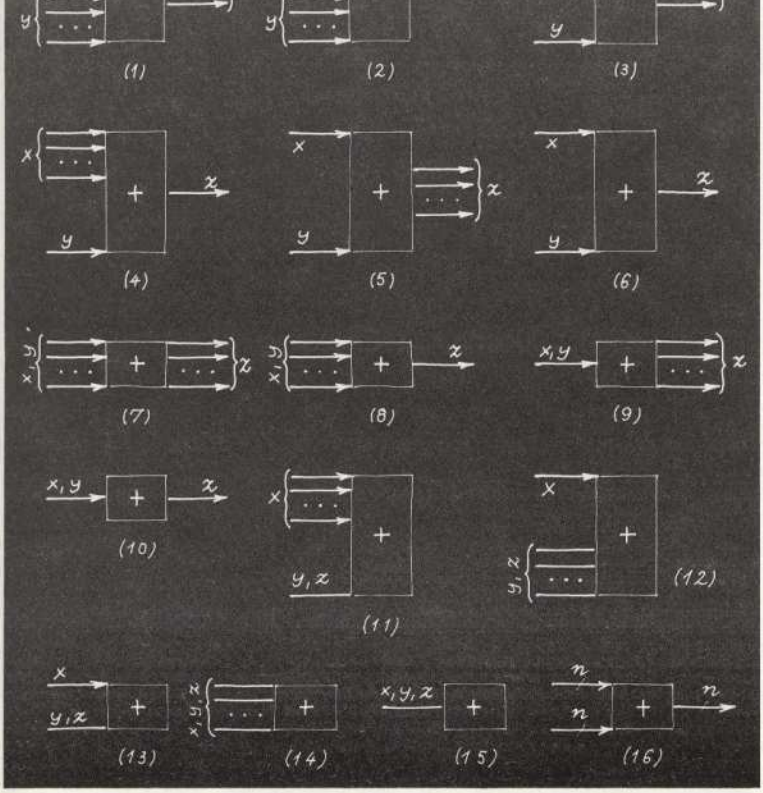
Példáinkból láthatóan, nagyon is sokféle, és mindegyik típus megvalósítható és meg is valósítják valamilyen gyakorlati hasznú berendezésben.

Mivel nekünk most nem célunk *minden lehetséges módon* digitális berendezést építeni, kiválasztunk egy – talán a leggyakoribb – típust és ezzel foglalkozunk, annak érdekében, hogy *egy lehetséges megoldást* előállítsunk. Típusunk a 2. ábra (1) példa típusa lesz.

Céljaink érdekében főleg – de nem kizárólag – olyan operátorokra lesz szükségünk, amelyek függvény-helyettesítési értéket képeznek. Ezek az operátorok tehát, mivel függvény-helyettesítésiértékkiszámító feladatot látnak el, függvény-helyettesítésiértékkiszámító egységek, függvény-helyettesítésiértékkiszámító gépek. Hogyan működjenek ezek a függvény-helyettesítésiértékképző (kiszámító) operátorok? Ezt most – a sok-sok lehetőség közül – mi határozzuk meg.

Rögtön az elején megmondjuk, hogy matematikai értelemben függvénnyel ritkán találkozunk a számológépben. Ritkán van, vagy talán *soha sincs egyidőben jelen a hozzárendelt azzal, amihez őt rendeljük.* Az „ős" (azaz a független változó értéke) *időben előbbi szokott lenni* a „kép"-nél (azaz a függő változó értékénél). Emiatt – függvények esetében is – mindig tudnunk kell valahonnan, hogy mit mely időpontban vegyünk számításba, mit mikor olvassunk le, figyeljünk meg, érzékeljünk stb. E függvénymodellezés lényegét jobban megértjük, ha transzformáció-végrehajtásról beszélünk, így rögtön világos, hogy az operátor által transzformációval képzett mindig később jelenik meg annál, amit transzformál. (Legfeljebb véletlen lehet a két egymáshoz tartozó értékpár egyidejű jelenléte a bemeneti, illetve kimeneti pontokon.)

2. ábra



Egyszerű kapcsolások, hálózatok

Most már nyugodtan rátérhetünk az operátorok kedvünk szerinti összekötögetésére és az így kialakított hálózatok vizsgálatára. A vizsgálatot most főleg egy szempontra korlátozzuk, arra, hogy a hálózat milyen transzformáció megvalósítására, milyen függvény modellezésére használható.

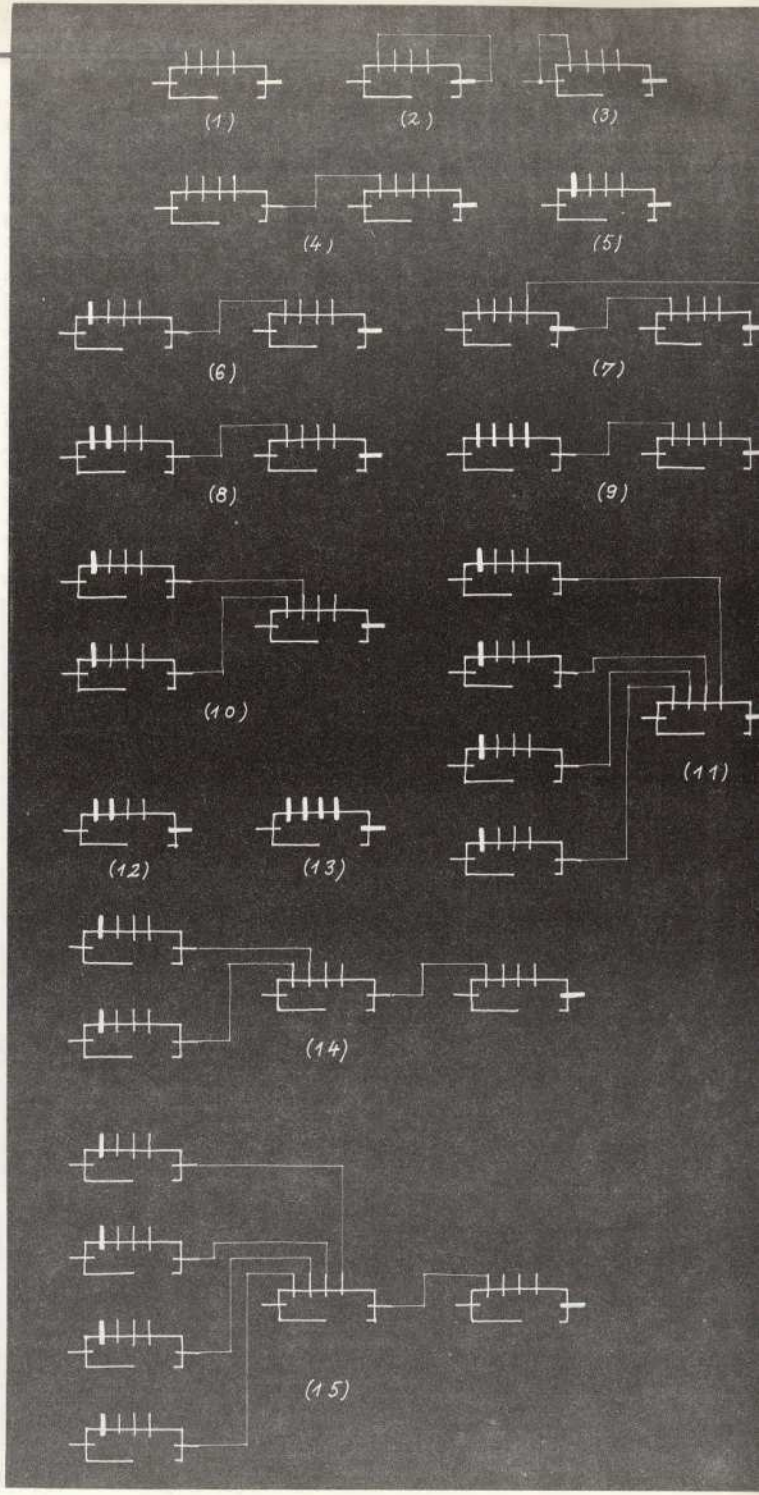
Mit jelent ez konkrétan? Azt, hogy ha a szóban forgó feladatot elvégző operátor bemenetein bizonyos ideje már fennállt egy állapot, akkor ennek az időtartamnak a kezdetét követő bizonyos idő múlva a bemeneteken levő állapotok és a rendszer által egyértelműen meghatározott (minden mástól viszont nem függő) állapot áll be, és marad fenn a kimeneteken, bizonyos ideig. (3 darab „bizonyos" idő szerepel a definícióban. Hogy ezek miktől és hogyan függenek pontosan, érdekes, de a gyakorlatban ritkán fontos kérdés, mert ezeknek az időknek mindentől független felső vagy alsó korlátjukkal szoktak tervezni, és ilyen korlátok a gyakorlati esetekben mindig lenni szoktak.)

Definíciónkban hangsúlyoztuk a kimeneti állapotoknak a rendszertől való függését. Ez a függőség nem azonos a rendszer előző állapotaitól való függőséggel. Ez is gyakori és sokszor hasznosított jelenség. Sőt a pontos „bizonyos idők" is függhetnek a rendszer előző állapotaitól és majdnem mindig függenek is, főleg más, például elektronikus rendszerek esetében. (Az ezektől is független felső és alsó korlátok azonban ilyenkor is léteznek.)

Példáinkban a rendszer előző állapotaitól való függést – most – mivel ez egzakt módon megindokolható, el fogjuk hanyagolni.

A kimeneti állapotok és a bemeneti állapotok között csak a kialakítótól a kialakított felé irányuló hatással számolunk. (Példáink többsége olyan, hogy

3. ábra



ez az irány egyértelműen megállapítható.) Az ellenkező irányú hatást elhanyagolhatónak vesszük, tehát feltételezzük a visszahatásmentességet. Természetesen ez is ritkán van így teljes pontossággal. A 3 bizonyos idő pontos meghatározásához visszahatással is számolni kellene, különösen más, például elektronikus operátorok esetében. Szerencsére igaz az is, hogy van olyan felső, illetve alsó korlátja a „bizonyos időknek" (bizonyos időtartamoknak), amely minden esetben igaz, és így a visszahatásmentesség feltételezése sem vezet helytelen eredményre, a felső és az alsó korlátokat használó becslésekkel dolgozó tervezés során.

A teljesség kedvéért említeni kell azt is, hogy a „bizonyos idők" egymástól és sok minden mástól is függhetnek. A gyakorlat által használt esetekben azonban erre is áll, hogy e függés oly kismértékű, hogy a már említett alsó és felső korlátok használatával minden problémát meg tudunk oldani.

Talán felesleges említeni, hogy rendszerünk viselkedése általában nem függ közvetlenül az időtől, csak az egyes jelenségek időbeli viszonyaitól; pneumatikus operátorunk például ugyanúgy működik hétfőn, mint kedden.

Kezdjük a kapcsolásokat, a hálózatépítést a legegyszerűbb esetekkel! A példákkal való ismerkedés során mindig célszerű a „mit csinál?", a „hogyan működik?" kérdés mellett azt is megvizsgálni, hogy a kapcsolás révén létrejövő rendszer kimenetén tapasztalható állapot, azaz az operátum mikor mit indikál (azaz mit jelez), mikor miről informál, mikor mire lehet belőle következtetni, mikor miről tudósít, hiszen a rendszerépítés egyik fő célja az informálás.

Nem minden kapcsolásban van szerepe minden bemeneti, illetve kimeneti pontnak (csőcsonknak); amelyeket az összetett rendszer be-, illetve kimeneteként használunk közülük, azokat vastagabb vonalszakasz jelöli a *3. ábrán*.

Az összes operátor táp-bemenetére természetesen egy táp-levegő vezeték csatlakozik. A rajzok áttekinthetősége érdekében a táplevegő-hálózat megrajzolását mellőztük.

A 3.(1) ábrán látható rendszer kimenetén mindig 1-et tapasztalunk, ez a rendszer állandó függvényt modellez. Kimeneti pontja egyenértékű a táplevegő-hálózat egy szabad csőcsonkjával. A 3.(3) és a 3.(4) rendszer ennek párja, szintén állandó függvényt modellez, melynek értéke azonban 0. E rendszerek kimeneti pontja a szabad légtérből jövő vezetékével egyenértékű. A 3.(5) ábra rendszere 0,5 értékre tükröző operátort modellez; 1-gyel tudósít a bemenet (valamennyi idővel előbbi) 0 állapotáról, és 0-val jelzi a bemenet (valamennyi idővel előbbi) 1 állapotát.

A 3.(6) rendszer az azonos leképezést modellezi, azaz 1-gyel indikálja a bemenet (valamennyi idővel előbbi) 1 állapotát, 0-val pedig (valamennyi idővel előbbi) 0 állapotát. Ez az operátor késleltetésre is használható.

A 3.(2) és a 3.(7) rendszer kemény dió. Most nem foglalkozunk velük. (Az Olvasó azonban szánjon rájuk néhány percet! Próbálja megfejteni, hogy hogyan működnek! Elemzésük később sorra fog kerülni.)

A 3.(8) rendszer 1-gyel indikálja bemenetei valamelyikén (valamennyi idővel előbb) fennállt 1 állapotot, 0-val pedig azt az esetet, amikor minden bemenete 0 állapotban volt. Az operandus-operátum vagy a bemeneti állapot–kimeneti állapot táblázat 2 bemenet esetére a következő.

operandus		operátum
0	0	0
0	1	1
1	0	1
1	1	1

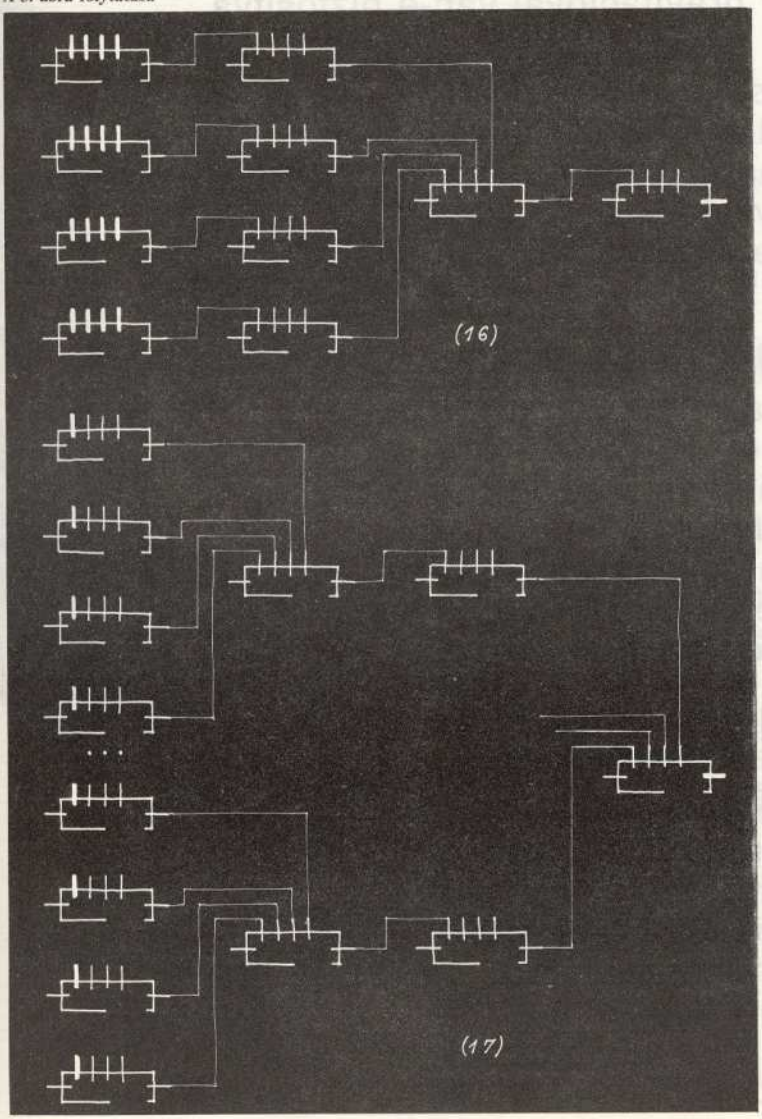
A 3.(9) rendszer ugyanolyan, mint a 3.(8), de 4 bemenete van. E rendszerek maximumfüggvény modellezésére használhatók.

A 3.(10) rendszer 1-gyel indikálja azt, hogy mindegyik bemenete (valamennyi idővel előbb) 1 állapotban volt, 0-val pedig az összes többi esetet. Minimumképző operációt végez, alsó burkoló függvény modellezésére használható.

A 3.(11) rendszer ugyanaz, mint a 3.(10) rendszer, de 4 bemenete van. Táblázata 2 bemenet esetére a következő.

operandus		operátum
0	0	0
0	1	0
1	0	0
1	1	1

A 3. ábra folytatása



A 3.(12) rendszer operandus–operátum táblázata 2 bemenet esetére:

operandus		operátum
0	0	1
0	1	0
1	0	0
1	1	0

Az operátum 1 voltából arra lehet következtetni, azaz a kimenet azt indikálja 1 értékkel, hogy mindkét bemenet (valamennyi idővel előbb) 0 állapotban volt. Minden más esetet 0-val indikál.

A 3.(13) ugyanaz, mint a 3.(12) de 4 bemenet esetére.

A 3.(14) rendszer azt indikálja 1 értékkel, hogy a bemenetek közül legalább egy (valamennyi idővel előbb) 0 állapotban volt; az egyetlen ettől eltérő esetet, amikor minden bemenet állapota 1 volt, 0-val indikálja. Táblázata a következő.

operandus		operátum
0	0	1
0	1	1
1	0	1
1	1	0

A 3.(15) rendszer ugyanolyan, mint a 3.(14), de 4 bemeneti pontja van.

A 3.(16) rendszer csak bemeneti pontjainak számában tér el a 3.(8) és a 3.(9) rendszertől.

A 3.(17) rendszer 8 bemeneti ponttal rendelkezik. Látható, hogy van hely még 2-szer négy bemeneti pont létesítésére. Ugyanazt a feladatot végzi, mint a 3.(10) vagy a 3.(11) rendszer.

A példák magyarázatánál – helykímélés céljából – nem említettük mind a három „bizonyos időt", csak egyet. Természetesen a másik kettőnek változatlanul szerepe van, a pontos működéstörvényben szerepelniük kell.

Természetesen vetődik fel a kérdés, hogy minden elképzelhető kapcsolásról meg tudjuk-e mondani, hogy minden elképzelhető bemeneti folyamat esetében hogyan fog működni. A válasz az, hogy nem. A gyakorlati esetekben sem tudunk válaszolni az ilyen kérdésekre, csak bizonyos tulajdonságú kapcsolások és bizonyos tulajdonságú bemeneti folyamatok esetében. Más a helyzet az elméletben. Ha van egy jó matematikai modellünk, azzal többre vagyunk képesek. A kérdés csak az, hogy emögött az (elméleti!) több mögött van-e valóság, és ha igen, milyen közel van hozzá. Nem tudjuk.

Felhívjuk a figyelmet arra, hogy egy jó matematikai modellnek már a csővezetékek alakját és hosszát is kezelnie kell, elektromos esetekben pedig például a kialakuló és állandóan változó elektromágneses teret is számításba kell vennie. Egyik követelmény sem tekinthető teljesítettnek. Szerencsére ezek a fájó tehetetlenségek a lehetséges esetek olyan tartományára esnek, amelyeket a gyakorlat ki tud kerülni.

Példáink meggyőzően bizonyítják, hogy pneumatikus operátorokkal kapcsolással nagyon sokféle operátor előállítható. A bemeneti pontok állapotáról nagyon sokféle információ létrehozására képes rendszereket tudunk felépíteni. Joggal vetődik fel a kérdés, hogy van-e ennek a tevékenységnek határa, azaz van-e olyan, csak a bemenetek állapotától függő 2 értékű (helyesebben 2 elemű értékkészletű) függvény, aminek helyettesítési értékét kiszámító rendszert már nem tudnánk operátorokból kapcsolással létrehozni. Könnyű bizonyítani, hogy ilyen függvény nincs. Azaz képesek vagyunk bármilyen operandus–operátum táblázathoz egy azt megvalósító rendszert megalkotni, a már megismert keretek között. Ezzel is foglalkozunk majd a következő alkalommal.

POGÁNY CSABA

Commodore 64 felhasználók figyelmébe ajánljuk az alábbi programokat:

1. *Kémény áramlástechnikai méretezés*
Alkalmas szén-, olaj-, gáztüzelésű kazánok kéményeinek méretezésére. A kémény anyaga lehet: tégla, beton, hőszigetelt fém.
Méretezhetők deflektoros kémények is, figyelembe vehető, hogy a kazánház légpótlása, ill. a kazán huzata mesterségesen, vagy gravitációval van-e biztosítva.

2. *Fűtőtest-kiválasztás*
Alkalmas központi fűtés tervezéséhez, hőszükséglet-számításra. A kiválasztott fűtőtestekről a mennyiségek összesítésével anyaglistát készít, mely sornyomtatón is lekérhető.

3. *TRACE-64 programozói segédlet*
Segédlet, mellyel az alkalmazói program elkészítési ideje a felére csökkenthető. Képes önmagát bármely területre áthelyezni, disassemblaláskor lehetőség van címkék használatára is. Leghatékonyabb eszköze a NYOMKÖVETÉS, zero-page területet nem bántja, s csak 4 kbájtot foglal el.

4. *SOFTKEEPER titkosító program*
Lehetővé teszi a futó programok, illetve programok által használt adatok tetszőleges kulcs szerinti átkódolását és titkosított formában történő tárolását. Az eredeti tartalomhoz csak a helyes kulcs ismeretében lehet hozzájutni.
A gépi kódú program a kódoló utasításon kívül, még 12 – a felhasználó munkáját megkönnyítő – utasítást ismer.

1. *Gépjármű nyilvántartó és járulékszámító program*

Alkalmas az 50/1981. (X. 31.) és a 85/1982. (XII. 4.) PM sz. rendelet alapján 20 termelőegységre 300 gépjárműnek a járulék számítása szempontjából szükséges adatainak tárolására, a járulék negyedévi és évvégi kiszámítására, valamint kinyomtatására.

2. *Nagyüzemi keresetadó számító programrendszer*

Alkalmas a mezőgazdasági és ipari vállalatoknál alkalmazható ún. nagyüzemi keresetadó kiszámítására, max. 10 000 dolgozó esetén. Sávonként összesíti a keresetet, az adót és a hozzá tartozó létszámot.

3. *Költségfelosztó programcsomag*

számlaszámra, kettő adatsor gyűjtésére, illetve tetszőleges nagyságú összeg felosztására, a felosztott összegnek a költséghely (számlaszám) sorrendje szerinti sorbarendezésére.

A programok megrendelhetők az alábbi címen:

Alkotó Ifjúság Egyesülés
Számítástechnikai Iroda
1519 Budapest, Pf.: 330.
Telefon: 112-666
113-608

Robottechnika

Korunk technikájában egyre nagyobb szerep jut a számítógépnek. Sokoldalúságával a mindennapi élet számos területén hódít teret magának. Fontos segítőtárs az adatfeldolgozásban, ügyvitelben, de újabban a tervezésben, a tudományos munkában is. Mi egy kevésbé ismert felhasználást mutatunk be: a számítógépeknek az iparban a munkafolyamatok irányítására való alkalmazását.

Régi törekvése az embernek, hogy mindennapi, fárasztó, esetleg egyhangú vagy nehéz munkáját könnyebbé tegye. Ezt a törekvést a történelem során az emberré válástól napjainkig végigkövethetjük az egyszerű munkaeszközök készítésétől kezdve a bonyolult ipari berendezésekig. A munkaeszköz fejlődése jelentősen befolyásolja a társadalom alakulását is. Míg az őskorban a szerszámok készítését a létminimum eléréséhez tette szükségessé, napjainkban már a gépek kiszorítják a munkásokat a gyárakból, az emberek a két kezi munkaterületekről a szellemi fejlesztő tevékenységek felé áramlanak. A gépek tehát átveszik fáradságos munkánkat: nekünk csak magukat a gépeket kell megterveznünk, szervezett rendszerbe állítanunk, működésüket irányítanunk, ellenőriznünk.

Ez utóbbi tevékenységeket korunkban már igen nehéz lenne emberekkel végeztetni, mivel ehhez az emberi képességeket meghaladó pontosság, gyorsaság, összetettség, összegangoltság kell. Ezeket az irányítási, ellenőrzési, vagyis szabályozási feladatokat oldja meg a szigorú kívánalmaknak megfelelően a számítógép.

Így ilyen folyamat részletes elemzését nem tűzhettük ki célunknak. Sokkal inkább azt, hogy nagy vonalakban bepillantást adjunk a számítógép gyakorlati felhasználásába. Először is készítenünk kell egy modellt, amit irányítani fogunk. A modellt úgy kell kiválasztani, hogy a valósághoz közel álljon, vagyis majdnem annyit tudjon, mint igazi társa.

Mire legyen jó a modell?

Egy modell általában akkor jó, ha alapvető tulajdonságaiban eredetijével megegyezik. Ez esetben nyilvánvalóan a felhasználás az alapvető tulajdonság, tehát: helyettesítse az embert valamilyen összetett, egyhangú munkában. Építsünk például egy robotot.

Mit tudjon a robot?

Elsősorban olyan műveleteket végezzen, amelyek sokszor ismétlődnek, olyan pontosságot, kitartást követelve az embertől, amit nem valószínű, hogy teljesíteni tudna. Könnyen lehessen átállítani egyik munkáról a másikra.

Egy modell általános tulajdonságaihoz természetesen az is hozzátartozik, hogy kiemeljen bizonyos részleteket az igazi rendszerből, és ezeknek a hangsúlyozott részleteknek alárendelje a többit. Jelen esetben a szabályozáshoz kapcsolódó részletek lesznek a legfontosabbak. Ennek megfelelően az egész konstrukciót úgy kell majd megalkotni, hogy a szabályozhatóságát lehetőleg minden egyes részegység segítse.

Ezek voltak a legfontosabb elméleti kérdések. Most már ismerjük az elvi célokat, következhet a megvalósítás.

Gyakorlati kérdések

Ha az emberi kezet akarjuk géppel helyettesíteni, mennyivel tudjon többet és mennyivel kevesebbet? Az emberi kéz (kar) elsődleges feladata tárgyak megfogása és mozgatása. A tervezés során figyelembe kell venni kezünk hiányosságait: elemezve a testek mozgását, majd megfigyelhetjük, hogy az emberi kéz a mozgatásoknak csak egy részét tudja elvégezni. Többek között ebben többet tud majd a robotunk. Karunk hiányosságait esetleg egy teljesen más felépítésű robotkarral pótolhatjuk.

A tárgyak megfogásának mechanizmusában viszont sok mindent elhagyhatunk, egyszerűsíthetünk. Nem lesz szükségünk az emberi kéz rendkívül összetett és bonyolult ujjaira, ujjperceire. Lehet, hogy robotkezünk nem tudja majd a tárgyakat annyiféleképpen megfogni, de erre nem is lesz szükség. Kevesebbet fog tudni a gyorsaság terén is, hiszen kezünk közelítőleg ugyanazt a pontosságot éri el a nagy és kis sebességű mozgatások esetén, robotkezünk pedig a pontosságot, a mozgások finomságát a legtöbb esetben a gyorsasággal fordított arányban tudja biztosítani.

A szabályozhatóságot szem előtt tartva, a kettő közül a gyorsaság rovására a pontosságot kell előtérbe helyezni. Ez viszont újabb előnyökhöz vezet az emberi kézzel szemben, hiszen a kéz pontosságának van egy felső határa, robotkezünknél pedig az egységnyi időre jutó elmozdulást lényegében korlátlanul csökkenthetjük, ami a pontosság hatalmas növekedéséhez vezet.

Milyen anyagok állnak rendelkezésre?

Sajnos ez a tényező minden előzetes számítást és tervezést elronthat. A mi modellünk viszonylag kevés alkatrészt igényel, inkább konstrukciós ötletekkel próbálhatunk bonyolult szerkezeti megoldásokat helyettesíteni. Ezenkívül rendelkezésünkre álló kész szerkezeti egységeket is felhasználhatunk.

A tervezés legfontosabb lépései

Az első tennivaló a testek mozgatásával kapcsolatos fogalmak tisztázása. *A test mozgása adottnak tekinthető, ha bármely pillanatban bármely pontjának meg tudjuk adni a helyét.*

Természetesen a bármely ponton lényeges pontot kell érteni, vagyis a testet is modelleznünk kell lényeges pontjai alapján. Ilyen testmodell többféle is van, de mi csak kettőt vizsgálunk:

1. a tömegpontot, amelyen olyan idealizált testet kell érteni, amelynek csak tömege van, kiterjedése elhanyagolható;
2. a merev testet, amely a mi szempontunkból a tömegpontnál is fontosabb. Ez olyan speciális pontrendszer, amely tömegpontjainak távolsága állandó.

A mozgások leírását nyugvónak tekintett környezetből az ún. vonatkoztatási rendszer segítségével végezzük. A tömegpont mozgásának leírásához a kinematikai leírásmódot alkalmazzuk. Ennek lényege, hogy a tömegpont helyének ismeretéből az idő függvényében következtetünk a sebességére és ebből a gyorsulására.

Egy pont helyét egy vonatkoztatási rendszerben a helyvektorral adhatjuk meg. A helyvektor egy kijelölt kezdőpontból a tömegpontba mutató vektor $\overrightarrow{OP} = \vec{r}$. Derékszögű koordinátarendszerben (amely három egymásra merőleges x,y,z tengelyekből áll) P tömegpont koordinátái P (x; y; z), az ún. gömbpolár koordinátarendszerben pedig P (r; θ; φ;) (*1. ábra*).

Látható, hogy háromdimenziós térben egy pont meghatározásához három egymástól független adat szükséges. A későbbiekben mutatunk alkalmazási példát az említetteken kívül másféle adattípusokra is.

A tömegpont pályáján azt a görbét értjük, amelyet a pont mozgása során leír. A pályát a pályaegyenlettel írhatjuk le. Ezt az említett három független változó (f,g,h) esetén a következő paraméteres egyenletrendszer adja:

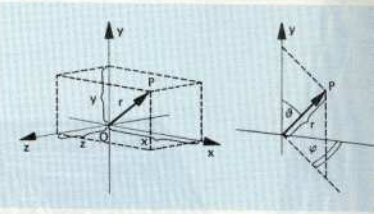
$$\begin{cases} f(t) = \dots \\ g(t) = \dots \\ h(t) = \dots \end{cases}$$

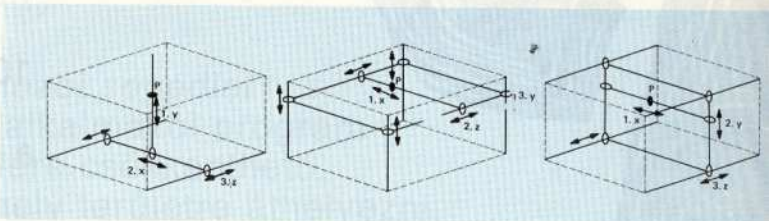
Ez a felismerés fontos tervezési irányvonalakat határoz meg. Tegyük fel ugyanis, hogy csak egy ilyen idealizált tömegpontszerű test mozgatására lesz szükség. Melyik legyen az a három adat, amellyel jellemezzük a pályáját? Nyilvánvalóan azt a hármat érdemes választani, amelyek egybeesnek a robot karjának mozgatóegységeit irányító adatokkal. Miért? Egyértelmű, hogy a robot karjának mozgatását is legalább három része lehet bontani, három, egymástól független mozgást végző részegységekre. Ezeknek a részegységeknek külön mozgatórendszerük van.

A szabályozhatóság elvét szem előtt tartva a legjobb az lenne, ha a szabályozórendszer és az előbb említett három mozgatórendszer közötti kapcsolat (vezérlés, visszajelzés) mentes lenne az adatok bizonyos előre meghatározott képletekkel való egymásba oda-vissza számolásától. Később ennek nagyon sok előnyét látjuk majd, főleg a mozgatott test pillanatnyi helyzetéről értesítő mérő-jelkibocsátó rendszerünk építésénél, hiszen ezt úgy alakíthatjuk ki, hogy nem a test, hanem a robotkar három részegységének pillanatnyi helyzetét méri.

A következőkben bemutatunk néhány példát a lehetséges három adat kiválasztására. Ezek – éppen az előbb leírtakból következően – már döntően befolyásolják a konstrukciót. Ezért a szerkezeti felépítés lehetőségeiről is döntenünk kell. Megvizsgálhatjuk e példák jó és rossz tulajdonságait, előnyeiket, hátrányaikat. Nem foglalkozunk egyelőre a működtető egységekkel; ezekhez a továbbiakban még eljutunk a kinematika részleteit ismertetve.

1. ábra





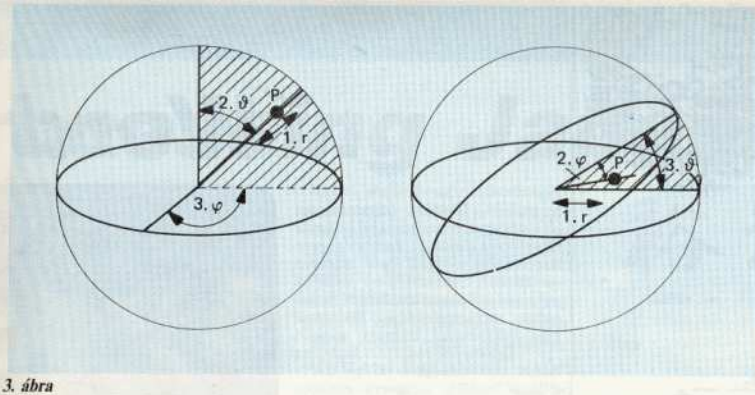
2. ábra

Fontosabb szempontjaink: a hozzáférhetőség a különböző helyzetű, egyelőre még mindig pontszerűnek tekintett testekhez, a megfelelő egyensúly problémája. Ezen azt értjük, hogy a robot karjának bármilyen helyzetében megőrizhesse egyensúlyát, megadott súlyintervelyzetű, egyelőre még mindig pontszerűnek tekintett testekhez, a megfelelő egyensúly problémája. Ezen azt értjük, hogy a robot karjának bármilyen helyzetében megőrizhesse egyensúlyát, megadott súlyintervallumú testek mozgatása közben. A mozgató szerkezeteknek esetünkben ugyanis nem az emelés, a súly megtartása az elsődleges feladata, hanem a mozgatás. Természetesen a test és a robotkar súlyának bizonyos részét a mozgatószerkezeteknek kell magukra vállalniuk. És most lássuk a példákat:

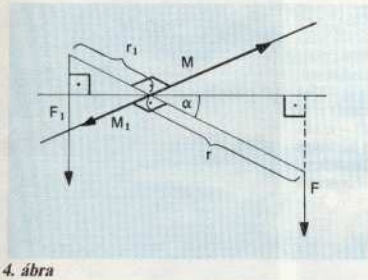
I. típus

A kiválasztott három adat legyen x,y,z (lásd az 1. ábrát). Ez azt jelenti, hogy mindhárom mozgatóegységnek egyenes vonalú mozgást kell megvalósítania. Nevezzük el ezeket a velük párhuzamos koordinátákról X,Y,Z-nek. A mozgatóegységek nyilvánvalóan egymáshoz kapcsolódnak, mégpedig úgy, hogy az első a tárgyat mozgatja, a második az elsőt, vele együtt természetesen a tárgyat, a harmadik pedig az előző kettőt.

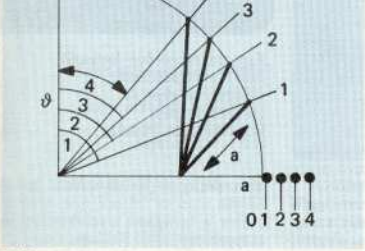
De mi legyen a sorrend? Ez attól függ, hogy milyen jellegű munkára van szükség. A hat lehetséges sorrend közül három különbözőt mutatunk be (*2. ábra*). Az ábrán bejelöltük a tér azon részét, amelyben aka-



3. ábra



4. ábra



5. ábra

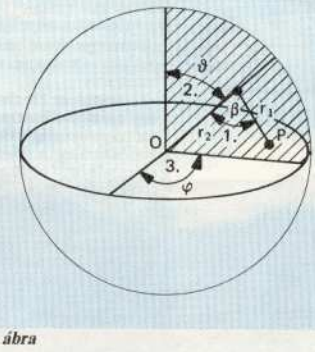
dály nélkül mozgathatja a testet a robot. Látható, hogy ez valamennyi esetben a fél tér, amit persze csökkentenek a méretezésből eredő hatások.

Például az a) esetben a Z irányú mozgatáshoz mindkét oldalán fel kell függeszteni az X irányú mozgatóegységet. Az egyensúly problémája ebben az esetben van a legjobban megoldva: mindenütt a vázszerkezetek viselik a test súlyát. Mégis az Y irányú mozgatásoknak is határt kell szabni. A Z irányú mozgásnál elképzelhető olyan szerkezeti megoldás, amelynél ez a mozgás végtelen hosszúságúvá tehető. Összegezve: a test mozgásának olyan tégla alakú térrész áll rendelkezésére, amelynek Z irányú kiterjedése végtelen.

A másik két esetben hasonló szempontokat figyelembe véve, az ábrán látható a térrész alakulása.

II. típus

A kiválasztott három adat legyen r,ϑ,φ (1. ábra). Az I. típussal ellentétben itt már csak két lehetőségünk van az R,ϑ,φ mozgások sorrendjére (*3. ábra*). A 2. árához hasonlóan itt is megjelöltük a test mozgásának határait. Az r irányú mozgáshoz stabilizálás szükséges, mivel az e mozgáshoz tartozó mozgatóegység csak egy ponton lehet felfüggesztve, hiszen a hozzá kapcsolódó φ vagy ϑ mozgatóegység forgómozgást végeztet vele. Az egyensúly létrehozására két mód kínálkozik. Az egyik: a forgástengely másik oldalán ellensúly segítségével létrehozni a két forgatónyomaték nagyságának egyenlőségét és irányának ellenkezőségét.



6. ábra

A forgatónyomaték (erőmomentum) $\vec{M} = \vec{r} \times \vec{F}$ melynek nagysága $|\vec{M}| = |\vec{r}| \cdot |\vec{F}| \sin \alpha$ (ez esetünkben ϑ vagy φ) ahol $\vec{r}$ az $\vec{F}$ támadáspontjába mutató vektor, $\vec{F}$ pedig $\frac{\Delta(m \cdot \vec{v})}{\Delta t}$-nek definiálható (*4. ábra*).

A két forgatónyomaték akkor lesz egyenlő, ha az $|\vec{r}| \cdot |\vec{F}|$ szorzat egyenlő lesz az $|\vec{r_1}| \cdot |\vec{F_1}|$ szorzattal. Az egyensúlyra vonatkozó paraméterek 1 indexüek. Mivel $\alpha_1 = 180° - \alpha$, tehát $\sin\alpha = \sin\alpha_1$.

Az ellentétes irány az $\vec{r}$ és $\vec{r_1}$ ellentétes előjeléből következik. Mivel m és m_1 állanók, csak r-től és r_1-től függ az egyensúly. Látszik, hogy ϑ és φ megváltozása nem befolyásolja az egyensúly kérdését.

A szerkezeti megvalósításnál érdemes figyelembe venni, hogy az $\frac{m}{m_1} = \frac{r_1}{r}$ összefüggés felhasználható arra, hogy az ellensúlyt ugyanazzal a meghajtó egységgel (R-rel) mozgassuk, mint a testet.

A másik lehetőség a rugós ellensúlyozás. Ez azonban nagyon bonyolult méretezést és kivitelezést igényel, ezért eltekintünk az ismertetésétől. Meg kell említenünk azonban, hogy kis súlyok esetén a vázszerkezet és a ϑ vagy φ mozgatóegységek megfelelő méretezésével az egyensúlyozó egység elhagyható. A mozgás határa – mivel r-nek bizonyos határon túlnövése esetén nem lehet az egyensúlyt biztosítani – megközelítőleg egy r sugarú gömbre korlátozódik. A „megközelítőleg" a gömb középső részén lévő rész elérhetetlenségéből és az egyes szerkezeti felépítéseknél fellépő, a teljes körbefordulást gátló tényezőkből ered. Meg kell jegyeznünk, hogy a valóságban ilyen II. típusok esetén általában megelégedhetünk egy félgömbnyi térrel is.

Vizsgáljuk most a II/a típust. A ϑ mozgatást rendszerint az *5. ábrán* látható módon szokták megoldani. Ezzel a következő problémák adódnak. 1. Korlátozott a szögelfordulási érték ($\theta < 180°$), de a legtöbb esetben ennél többre nem is szokott szükség lenni. 2. Át kell számítanunk az ábrán jelölt „a" hosszúsági értéket a szögelfordulási értékekre, mivel ezek nincsenek egyenes arányban egymással.

Például az egyenletes sebességű mozgásnál a mozgatóegységeknek változó sebességű mozgást kell végezniük. Ez a megoldás azért vált be, mert ebben az esetben olyan mozgatóegységeket is lehet forgómozgás létrehozására használni, amelyek csak egyenesvonalú mozgást tudtak volna megvalósítani.

III. típus

Ezeknek a robotoknak három szögelfordulási adatra van szükségük. Legjellegzetesebb típusuk a *6. ábrán* látható. Mozgása az $r_1 + r_2$ sugarú gömbre korlátozódik $r_2 = r_1$ esetén. Ha $r_2 \neq r_1$, akkor egy $|r_2 - r_1|$ sugarú gömbbel kevesebb. Természetesen érvényesek erre a típusra a II. típushoz tartozó mozgáskorlátoknál tett megjegyzések is.

A bemutatott típusoknak sok keveréke ismert, mi azonban csak ezeknek a kivitelezésével fogunk foglalkozni.

Az eddigiekben talán túl sok elmélettel terheltük meg olvasóinkat, ezért most néhány gyakorlati kérdéssel foglalkozunk.

Fontos kérdés, hogy robotkarunk mozgatásához milyen erőforrásokat használunk. Jó megoldás hidraulikus meghajtóművek alkalmazása, de ilyenek nekünk sajnos nem állnak rendelkezésünkre. Maradnak tehát az elektromotorok.

Ezeknek két, számunkra fontos fajtájuk van: a szinkronmotorok és az ún. léptetőmotorok.

A szinkronmotorokkal gyakran találkozhatunk, hiszen ezeket alkalmazzák a kis villanyvasutakban, motoros modellekben. Előnyük, hogy viszonylag kis méretűek, és működtetésükhöz mindössze két vezetéket kell elvezetni hozzájuk. Nagy hátrányuk viszont viszonylagos pontatlanságuk. Emiatt és nagy fordulatszámuk miatt csak nagy áttétel segítségével lehet pontos mozgások megvalósítására használni ezeket.

Léptetőmotorokkal a hétköznapi életben ritkán találkozhatunk, gyakran előfordulnak viszont például nyomtatók, rajzolók meghajtóműveként. Az a tulajdonságuk, nevezetesen a pontosság, amely miatt jól alkalmazhatók ilyen célokra, alkalmassá teszi őket robotkarok meghajtására is. Ezek a motorok ugyanis egy fordulatot kis lépésekre bontva végeznek el, innen származik a nevük. Az egy lépésre eső szögelfordulás típusonként változik, általában 1 és 3 fok közötti érték.

A léptetőmotorok működési elvükből következően nagyon pontosak, és sebességüket is tetszőlegesen változtathatjuk, mivel ez csak a lépések gyakoriságától függ. Természetesen van azért egy maximális követési időköz. E motorok hátránya viszont, hogy a legkisebb típus is sokszorta nagyobb, mint egy normál szinkronmotor, továbbá, hogy vezérlésükhöz típustól függően legalább 6 vezeték szükséges.

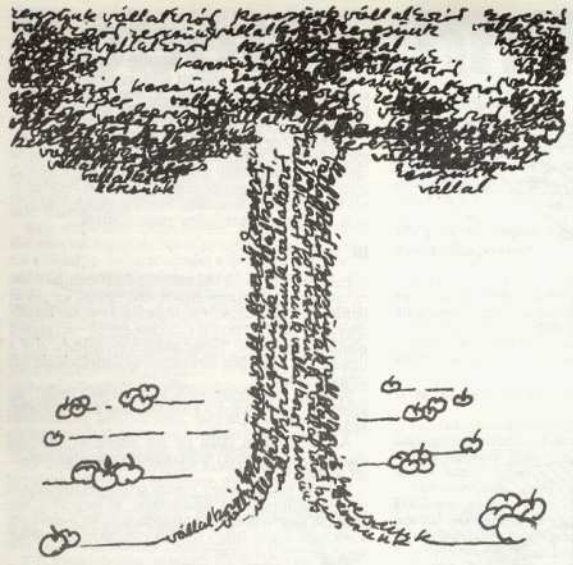
Mint már említettük, a robotok fő alkalmazási területe az ipar. Az ipari robotokra általában jellemző, hogy nagyon rugalmasak, könnyen átprogramozhatók egyik munkáról a másikra. Visszacsatolási folyamat nélkül, tárolt programjuk szerint végzik munkájukat. Különféle típusaikat ma már nagyon sok cég gyártja, és alkalmazási területük is széles. Többségük az útvonal egyszeri bejáratásával programozható. Áruk 20–30 ezer dollár között váltakozik.

Az első ilyen robotok gyártóinak egyike, az Unimation cég így mutatja be Unimate nevű robotját: „Itt van a fémipar új munkásának típusa: nem tart kávészünetet, huszonnégy órás munka alatt sem fárad el, nem érdekli a kereset és a nyugdíj. Érzéketlen a porra, füstre, melegre. Ez az ipari robot".

A másik fejlesztési út a szinkron-manipulátorok kifejlesztése. Ilyen a H. Kleinwächter intézetében kidolgozott Syintelmann elnevezésű manipulátor, amely egy irányító személy mozdulatait távolabb, vele egyidőben, nagy potosággal utánozza.

Ezeknek a fejlesztéseknek távoli célja az ún. szuperintelligens robotegységek kifejlesztése, melyek hűen reprodukálhatnák az ember mozgását, és a működésük közben adódó szituációknak megfelelően bizonyos döntésekre is képesek lennének. De létrehozásuk a távoli jövő feladata. Ugyancsak várat még magára az emberi járást utánzó szerkezetek elkészítése. Léteznek ugyan már ilyenek, de ezek is csak egy irányító személy mozgását képesek utánozni.

HERENDI MIKLÓS



Vállalkozót keresünk

SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÓK KATALÓGUSÁ-nak

egyszeri összeállítására és folyamatos karbantartására

A katalógus nyújtson egyszerű tájékozódási lehetőséget a számítástechnikai tevékenységi körökhöz rendelten

- **a szolgáltatók által eddig elvégzett tevékenységről,**
- **a referenciahelyekről,**
- **az értékesíthető, ill. az adaptálható munkákról,**
- **a kapacitásadatokról, ill. a vállalkozási készségről.**

A katalógus összeállítására vonatkozó ajánlatokat, a tartalmi, formai kivitelre és a vállalkozás körülményeire vonatkozó elképzeléseket kérjük a
1502 Budapest, Pf. 129
címre beküldeni.

ÉLELMISZERIPARI ÜGYVITELSZERVEZÉSI ÉS GÉPI ADATFELDOLGOZÓ VÁLLALAT

Telex: 22-6035

TÉNYLEG „KÍNAIUL V

Word

A számítástechnika alkalmazásának nagyon sok olyan vonatkozása van, ami az emberi kultúra és írásbeliség nemzeti sajátosságaihoz kapcsolódik. Különösen igaz ez olyan ország esetében, mint Japán. Cikkünk szerzője hosszabb időt töltött a szigetországban ösztöndíjas tanulmányúton. Úgy gondoljuk, igen tanulságosak a japán írásrendszer számítógépes vonatkozásaival kapcsolatos tapasztalatai.

„Nekem ez kínaiul van" – szoktuk mondani Európában, ha valami nagyon érthetetlen számunkra. Valószínűleg a Nanzan Egyetem számítóközpontjában tanuló külföldi diáktársaim egy része is így kiáltott volna fel a számítógépterminálokon megjelenő sok kínai írásjel láttán, ha ezek a diákok történetesen nem a környező délkelet-ázsiai országok valamelyikének kínai kolóniáiból érkeztek volna a japán egyetemre.

Hasonló szövegeket ugyanis gyakran láthattunk a terminálokon, tekintve, hogy az elmúlt években Japánban is rendkívül gyors fejlődésnek indult a számítógépes szövegfeldolgozás, és ezen belül a szófeldolgozás. Hasonlóan az USA-hoz, ahol ez a terület a számítógép-alkalmazások legdinamikusabban fejlődő területévé nőtte ki magát néhány évvel ezelőtt.

A Japánban „word processing"-nek nevezett szófeldolgozás azonban tartalmilag nem azonos azzal, amit az USA-ban értenek ezen a terminológián. Japánban ugyanis jelenleg háromféle írásrendszert használnak: a kandzsi, a hiragana és a katakana rendszereket.

Az előbbi írásrendszer elemei önálló jelentéstartalommal rendelkező, kínai eredetű írásjelek, az utóbbi kettő elemei pedig a japán nyelvben használt szótagok fonetikus írásjelei. A kandzsi jelkészlet elemeinek száma több tízezer, amiből 1850-et tanítanak meg a kötelező kilencéves oktatás során. A hiragana és a katakana rendszer alapjeleinek száma 50.

A normál japán szövegekben mindhárom írásrendszert egyforma gyakorisággal használják. Az önálló jelentéstartalommal rendelkező szavakat (tehát az igéket, a főneveket, a mellékneveket stb.) kandzsikkal jelölik; a ragokat, jeleket, képzőket hiraganákkal; az idegen szavak fonetikus átírására pedig a katakanákat használják.

Egy széles körben használt japán wordprocessor, a korszerű gépírás elengedhetetlen számítógépes eszköze

Bár a hiragana nyelvkészlet 50 alapjelével (tehát 50 alapszótaggal) minden japán szöveg leírható, a gyakorlatban ezt az egyszerű írásmódot mégsem használják. A kandzsikkal ugyanis tömörebben és érthetőbben tudják kifejezni a gondolataikat. Igaz ugyan, hogy ennek megértéséhez sokszor a japán – és esetenként a kínai – kultúrtörténet ismeretére is szükség van.

A 日 kandzsi például eredeti formájában kör alakú volt évezredekkel ezelőtt, csak később vált szögletessé. Így érthető, hogy az írásjel jelentése: *nap*. A 木 kandzsi jelentése *fa*, a 林 kandzsié *liget*, a 森 kandzsié pedig *erdő*.

Ha a *fa* alsó részét kiemelem, akkor a 本 kandzsit kapom, amelynek a jelentése: *gyökér* vagy *eredet*. Ha a *fát* és a *napot* kombinálom, egy új kandzsit kapok, aminek a képe a következő: 昇. Ennek jelentése: *kelet*, hiszen ke-

rocessing Japánban

Amit sohasem tud pótolni a számítógép, az a művészi kalligráfia vonalvezetésének sajátos karaktere

Ezek mellett az érthetőséget megkönnyítő előnyök mellett a kandzsi írásrendszernek egyéb előnyei is vannak. A japán nyelv sajátossága, hogy nagyon sok azonos kiejtésű, de különböző jelentéstartalmú szót tartalmaz. Ezeket a jelentésbeli különbségeket tehát csak a kandzsi írásjelekkel leírt szövegekből lehet felismerni.

Az egyik leggyakoribb japán személynévnek, a „Hirosi"-nak a jelentése például akár ötféle is lehet, attól függően, hogy milyen kandzsit használunk a szó leírására (lásd az *1. ábrát*). Ha ezt a szót latin betűs átírásban vagy a fonetikus szótagírások egyikével, például a hiraganával írnánk ひろし (kiejtése: hi) (kiejtése: ro) (kiejtése: si) alakban, akkor ezek a kandzsival kifejezhető jelentésbeli különbségek mind érthetetlenné válnának.

1. ábra

2. ábra

きょ	ねん	の	く	がつ	に	に	ほん	へ	い	きました	。
去	年	の	九	月	に	日	本	へ	行	きました	。
múlt	év	nek a	9.	havá	ban	Japán		ba	utaz	tam	.

A múlt év szeptemberében Japánba utaztam.

leten látható ilyen jelenség, hogy a nap jön fel az égre egy fa mögött.

Ha a 日 és a 本 kandzsikat vonom össze egymással, akkor a 日本 jelkombinációt kapom, aminek a jelentése: *Japán*. Japán ugyanis a „Nap Eredetű Ország"; a legendák szerint az első császár Ameteraszu Omikaminak, a Napistennőnek volt az unokája.

A 京 kandzsi jelentése *város*, esetleg *főváros*. Ennek alapján érthető, hogy 東京 *Tokiót* jelenti, szó szerint *Keleti Fővárost*, mivel Japán három állandó fővárosa közül ez van a legkeletebbre.

Ha már most a „kelet" írásjelét ismerjük, akkor tanuljunk meg még két másik égtájat is: 北 jelentése *észak*, 南 jelentése pedig *dél*. Mivel Kínában ugyanezt az írásrendszert használják, ezután már azt is megértjük, hogy Peking írásjelei 北京 *Északi Főváros*t, Nankingéi pedig 南京 *Déli Főváros*t jelentenek.

A fentiekből tehát kitűnik, hogy az ideogramma írásmódnak több előnye is van a betűkből felépített írásrendszerekhez képest. Hátránya viszont az, hogy a több ezer jelet 10–15 évi tanulással lehet csak elsajátítani, az írógéppel való írás pedig a több ezer jel miatt hihetetlenül lassú és nehézkes. Ez utóbbi probléma megoldására fejlesztették ki a japán wordprocessort.

Ennek a szoftverrendszernek a feladatköre azonban – amint erre már utaltam – egyáltalán nem azonos az amerikai értelmezés szerinti wordprocessor feladatkörével. Tekintve, hogy a japán szövegekben sohasem jelölik a szóközöket, hanem folyamatosan írják a jeleket egymás után, a nyomdakész szövegek hasábformátumának kialakítására például egyáltalán nem kell gondot fordítani.

A rendszer lényege abban van, hogy a szöveg írásakor a fonetikus szótagírás-jeleket (tehát a csupán 50 alapjelből álló hiraganákat vagy katakanákat) billentyűzzük be a gépbe, majd ennek alapján a rendszer megkeresi a számítógép tárolójában tárolt több ezer ideogramma közül az adott jelsorozathoz tartozó kandzsikat. Azokat ezután mind kiírja a képernyőn, ahonnan könnyen kiválasztható a megfelelő, és ez ezután bekerül majd a végleges szövegbe. Gondoljunk például a már említett Hirosi esetére.

Bármilyen hihetetlennek is tűnik, egy ilyen wordprocessor alkalmazásával gyorsabban elő lehet állítani a háromféle írásrendszer jeleivel egy nyomdakész japán nyelvű szöveget, mint ha például angol fonetika szerint, latin betűs írógépen gépelnék le. A Hiroshi név latin betűs írásmódja esetén például 9 billentyűt kell leütni az írógépen: egy betűváltás a nagy kezdőbetű miatt, hét betű és egy szóköz – a japán wordprocessoros írásmód esetén pedig csak ötöt: három szótagbillentyű: てらし, egy karaktermegjelenítő billentyű és egy billentyű, amely jelzi, hogy a képernyőn megjelent karakterek (kandzsik) közül melyiket kell kiválasztani.

Befejezésül leírok egy teljes japán mondatot a hiragana jelekkel, ahogy azokat a wordprocessorokba betáplálják. Utána meg adom azt a mondatot, ahogyan az nyomtatásban megjelenik (kandzsi és hiragana speciálisan ragokra), majd alájuk írom az egyes jelek magyar megfelelőjét *(2. ábra)*. A mondat fordítása: A múlt év szeptemberében Japánba utaztam.

Ebből a példamondatból látható, hogy a japán nyelvű mondat leírásához csupán 27 leütésre van szükség a japán wordprocessoron, amelynek klaviatúrája alig nagyobb, mint a Magyarországon használatos írógépeké, hiszen lényegében csak az 50 alapjel van rajta. A mondat magyar megfelelője csak 42 leütéssel írható le.

A példamondatból azonban más is kiderül. Nevezetesen az, hogy a japán és a magyar mondat szerkezete nagyon hasonló. A szórend teljesen azonos; egyik nyelvben sincsenek prepozíciók, csak jelek, ragok és képzők stb. Persze ez a jelenség nem is igazán meglepő, ha belegondolunk, hogy mindkét nyelv az urál-altaji nyelvek közé tartozik.

A két nyelv szerkezetének összehasonlító elemzése túlmutatna ennek a cikknek a keretein. Az azonban talán már az eddigiekből is kiderült, hogy nekünk, magyaroknak a japán mégsem egészen „kínaiul van".

MÁRKUS GÁBOR

PROGRAMOK

COMMODORE VC–20
Finom felbontású grafika

Annak ellenére, hogy a VC–20 BASIC nyelvjárása semmilyen rajzoló utasítást nem tartalmaz, mégis lehet vele némi ügyeskedéssel rajzolni. Két példán mutatom be, hogyan oldható ez meg BASIC programból.

Rajzolás botkormány segítségével

Az *1. programot* betöltés után RUN-nal indítsuk el. A képmező mérete kisebb lesz, megváltozik a keret színe, megjelenik egy piros pont a képmező közepén. Ebből kiindulva a botkormány mozgatásával vonalakat rajzolhatunk nyolcféle irányban. Ha megnyomjuk a lövőgombot, a rajzolópont láthatatlanul elmozgatható. Így szaggatott vonalat is tudunk rajzolni, és törölni is tudjuk a már ábrázoltakat.

Ha megunjuk a rajzolást, nyomjuk meg a SPACE billentyűt. Ekkor a képernyő ismét alapállapotba kerül, és egy kérdőjel jelenik meg rajta. A program tőlünk vár utasítást arra, hogy mit csináljon.

Válaszolhatunk az 1, 2, 3 billentyűk lenyomásával. Az első esetben törli az előbb rajzolt ábrát, és elölről kezdhetjük a rajzolást. A második esetben visszakapjuk az előzőleg készített rajzot, melyet ott folytathatunk, ahol az előbb abbahagytuk. A harmadik esetben a programból való kilépésre utasíthatjuk gépünket.

Gépi kódú monitorprogram segítségével a rajzolt ábrát ki tudjuk menteni mágneses tárolóra, így nem vész el a rajz a számítógép kikapcsolásakor.

A program készítése

Ilyen program készítéséhez ismernünk kell a VC–20 memóriatérképét. Észrevehetjük, hogy a $1DFF címtől lefelé a sztringváltozók és maguk a sztringek részére, továbbá a tömbök részére fenntartott helyek eredeti funkciójukban nem kellenek a rajzoláshoz, tehát nyugodtan igénybe vehetjük ezeket azzal a megszorítással, hogy nem íratunk ki betűket a képernyőre, nem végzünk semmilyen sztringműveletet és nem használunk a programban tömböket.

A képernyőt apró cellákra kell osztanunk. A cellákból vízszintesen 16 darabot, függőlegesen 10 darabot kell elhelyeznünk ahhoz, hogy megközelítően négyzet alakot kapjunk, mivel egy cella maga is egy 8 × 16 pontból álló mátrix. Így egy 128 × 160 pontból, azaz 160 cellából álló képmezőt kapunk.

Ehhez a képernyőtár címeinek értékét is át kell szervezni, mégpedig a kezdő címtől ($1E00) egészen a $1E95 címig.

A fentebb említett sztring- és tömbtárolók címeinek tartalmát is törölni kell ($1400–$1DFF) indulásnál és akkor, ha új rajzot kívánunk készíteni.

A botkormány értékeinek lekérdezése sokkal bonyolultabb, mint a Commodore 64-nél. Az alábbiak szerint kell eljárnunk:

1. Át kell állítani a bemenetet ellenőrző tárrészben a $9122 rekesz értékét 127-re, hogy a botkormány vezérlőutasításait a gép fogadja.

2. Állandóan le kell kérdeznünk a $911F és $912Ø tárrekeszek értékeit, ugyanis ezeket állítja át a botkormány. Pontosabban a $911F tárrekesz 2. bitjét a felfelé mozgatás, 3. bitjét a lefelé mozgatás, 4. bitjét a balra mozgatás, 5. bitjét a lövőgomb, a $912Ø rekesz 7. bitjét a jobbra mozgatás állítja át.

bitek	7	6	5	4	3	2	1	0
$911F	–	–	lövés	balra	le	fel	–	–
$912Ø	jobbra	–	–	–	–	–	–	–

A billentyűzet lekérdezését a $C5 tárrekesz végzi. Ezt használjuk fel a SPACE billentyű figyeléséhez.

A képmezőből nem szabad a botkormányt vezérelve kilépnünk, mert ezeket a képmezőértékeket a program a már többször említett RAM területen tárolja. Amennyiben ebből kilépnénk, lefelé belecsorognánk a változó-, majd a BASIC program tárolóba, illetve felfelé a képernyőtárba, és ezzel tönkretennénk programunkat.

Ezt a hibát megakadályozhatjuk, ha figyeljük X és Y értékét, és nem engedjük azokat Ø alá és 126, illetve 158 fölé emelkedni.

A működés

20. sor: Letiltja a RUN–STOP és RESTORE billentyűk működését, mivel használatuk szépséghibát okoz

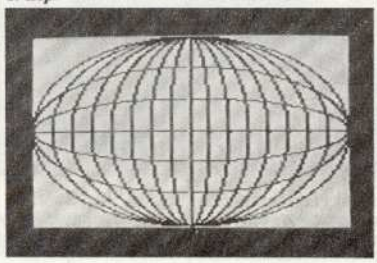
30. sor: Beállítja a képmező méretét és a keret színét

40. sor: Megjeleníti a képernyőtár tartalmát

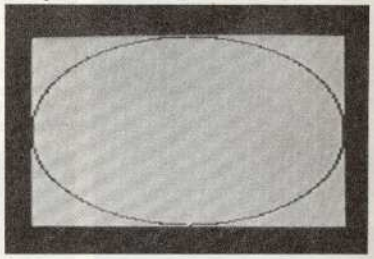
50. sor: Lehetővé teszi a botkormány állapotának beadását

70. sor: A lefoglalt tár törlését végzi és a kezdőpontot középre állítja

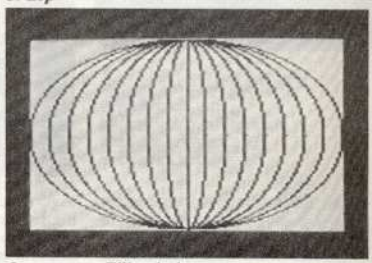
1. kép



2. kép



3. kép



1. program. Rajzolás botkormány segítségével

```
10 T=5120:O=7663:R=36864:K=37151
20 POKE788,194:POKEK-1,3
30 PRINT"■":POKER,17:POKER+1,45:POKER+2,144:POKER+3,21:POKER+5,253:POKER+15,39
40 FORA=1TO16:FORS=1TO10:POKEO+S*16+A,A*10+S-11:NEXT:NEXT
50 POKEK+3,127
60 IF Z=2THEN80
70 FORI=TTOO+16:POKEI,0:NEXT:X=64:Y=80
80 G=PEEK(K):X=X+((GAND16)=0):Y=Y+((GAND4)=0):Y=Y-((GAND8)=0)
90 IFPEEK(197)=32THEN180
100 X=X-((PEEK(K+1)AND128)=0):Q=-((GAND32)=0)
110 IFX<1THENX=1:GOTO100
120 IFX>126THENX=126:GOTO100
130 IFY<1THENY=1:GOTO100
140 IFY>158THENY=158:GOTO100
150 IFQ=1THEN170
160 W=T+INT(X/8)*160+Y:POKEW,PEEK(W)OR2↑(7-XAND7):GOTO80
170 W=T+INT(X/8)*160+Y:POKEW,PEEK(W)AND255-(2↑(7-XAND7)):GOTO80
180 POKER,12:POKER+1,38:POKER+2,150:POKER+3,174:POKER+5,240:POKER+15,27:POKEK+3,
255
190 PRINT"■":INPUTZ:IFZ=1ORZ=2THEN30
200 IFZ=3THENEND
210 GOTO190
READY.
```

2. program. Ellipszisek

```
10 T=5120:O=7680:R=36864:K=30720
20 PRINT"■":POKER,19:POKER+1,50:
   POKER+2,144:POKER+3,32
30 POKER+5,253:POKER+15,26:POKE56,28
40 FORA=0TO15:FORS=0TO15
50 POKEO+S*16+A,A*16+S
60 POKEO+S*16+A+K,0
70 NEXT:NEXT
80 FORZ=TTO7167:POKEZ,0:NEXT
90 FORI=64TO0STEP-8
100 FORG=0TO2*▲STEP▲/200
110 C=SIN(G)*I+64:D=COS(G)*63+63
120 X=INT(C):Y=INT(D)
130 W=T+INT(X/8)*128+127-D
140 POKEW,PEEK(W)OR2↑(7-(CAND7))
150 NEXT:NEXT
160 FORP=0TO42STEP21
170 FORG=0TO2*▲STEP▲/200
180 C=SIN(G)*64+64:D=COS(G)*P+64
190 X=INT(C):Y=INT(D)
200 W=T+INT(X/8)*128+127-D
210 POKEW,PEEK(W)OR2↑(7-(CAND7))
220 NEXT:NEXT
230 GOTO 230
READY.
```

80. sor: Átszámítja a botkormány-bemenet értékeit X és Y koordinátákká
90. sor: A SPACE billentyű lenyomását figyeli
100. sor: Lehetővé teszi a jobbra történő mozgást és lekérdezi a lövőbillentyű állapotát
110–140. sorok: Meggátolják a képmezőből való kilépést
150. sor: A lövőbillentyű állapotától függően a megfelelő sorra (rajzoló vagy törlő) ugrat
160. sor: Kiszámítja a pont helyét és ábrázolja azt
170. sor: Kiszámítja a pont helyét és törli, illetve nem ábrázolja azt
180. sor: Alaphelyzetbe állítja a képernyőt

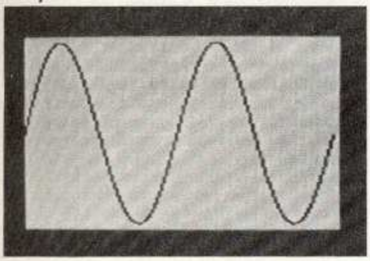
Ellipszisek

A *2. program* előre meghatározott függvényt rajzol, jelen esetben különféle méretű ellipsziseket. *(1. kép)*.

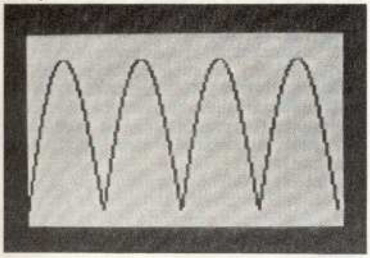
A program készítése

Ez a program az első program változata. Képmezejének alakja hosszabb az előzőnél. Pontosan 2 kbájtnyi tárat szabadítunk fel a $1400 címtől, és ez lehetővé teszi, hogy egy 16 × 16-os mátrixot alakítsunk ki a képernyőn. Miután a felszabadított tár nagysága szabja meg a képmező nagyságát, itt is az előző program szerint kell eljárnunk, azaz be kell szűkíteni a képmező határait a VIC kontrollregiszter bitjeinek megváltoztatásával.

4. kép



5. kép



A működés

20–30. sorok: Megváltoztatják a képmező méretét és a keret színét
40–70. sorok: Megjelenítik a képernyőtár tartalmát
80. sor: Törli a lefoglalt tárat
90. sor: Az ellipszis nagy tengelyének méretét állítja át kisebbre
110–120. sorok: Az ellipszis pontjainak helyét számolják ki
130–140. sorok: Kirajzolják a pontokat
160. sor: Az ellipszis kis tengelyének méretét állítja nagyobbra (Lásd *2., 3. kép.*)

A programot lerövidítve, a 110. sorban D után más függvényeket, illetve értékeket írva, különféle ábrák rajzolhatók.

100 FORG = 0TO4*Π STEP Π/200
110 C = 10*G:D = SIN(G)*60 + 63 (Lásd *4. kép.*)
100 FORG = 0TO4*Π STEP Π/400
110 C = 10*G : D = ABS(SIN(G))*100 + 10 (Lásd *5. kép.*)

SZILÁGYI GÁBOR

COMMODORE 64

A lemezkapacitás növelése

Az ügyvitelszervezési, ipari és kereskedelmi alkalmazások után a mezőgazdaságba is betörtek a személyi számítógépek. Különösen kedvelt e területen a Commodore 64, mert viszonylag olcsó, sokrétűen felhasználható és könnyen kezelhető. A hazai számítástechnikai gyakorlat azonban ettől a géptől is többet vár el, mint amire tervezték. A programozó pedig – hallván az igényeket – vagy széttárja a karját és zavaros magyarázatba kezd, hogy mit miért nem lehet, vagy beleássa magát az operációs rendszerbe, és ravasz trükköket eszel ki a kívánság teljesítésére, általában a kapacitás vagy a műveletvégzés sebességének növelésével.

Így jártunk mi is, akiknek az Állattenyésztési és Takarmányozási Kutató Központban nagy létszámú állatállományok adatait feldolgozó programot kellett írnunk Commodore 64-re. A 95 mezőből álló, 1500–2500 rekord tárolása és közvetlen elérése a lemezen számunkra is komoly fejtörést okozott. Szoftvermegoldást kellett találnunk az adatok sűrítettebb tárolására.

Először a lemez eredeti adatkezelését vizsgáltuk. Kiderült, hogy a rendszer a szám típusú változókat ugyanúgy, mint a betűket, ASCII kódban, karakteresen tárolja, külön bájtban ábrázolva a szám előjelét, a szám után pedig egy blanket és egy mezőelválasztó jelet ír, amely a többi számtól való elkülönítésre szolgál. Ezt az elhelyezési módot túl „szellősnek" találtuk, ezért egymástól függetlenül, két módozatát is kidolgoztuk a rekord tömörítettebb tárolásának.

1. program

```
10 DIM A(3),B(2):                    REM MEZOHOSSZ TOMB,KONVERTALT ADATOK TOMBJE
15 :
20 REM *****************    KONVERZIOS TABLA STRINGBEN    ******************
25 :
30 SF$="0123456789O+-£QWERTYUIOP@*↑ASDFGHJKL=ZXCVBM"
40 SF$=SF$+"./!#$%&'()+|◤●○‾⌐|╮┌┐└┼♠♥‾⌐||╲└┘]♦+‾"
50 SF$=SF$+"×|╱╲◯?▒▒H┼‾‾▬▬◥┌▬|||‡↑└ ▔▚||"
52 :
55 B(0)=12:B(1)=145:B(2)=5000:      REM ********    A MINTAPELDA      *********
58 :
60 REM ************    MEZOHOSSZAK BEOLVASASA "A,B" TOMBBE    **************
65 PRINT"⅃▓▓▓▓";"A  PROGRAM AKKOR JO, HA A 12,145,5000   SZAMOKAT IRJA KI"
70 FOR N=0 TO 3:READ A(N):NEXT N
80 :
400 REM  **********   BINARISBOL 121-BE MINDEN MEZOT KONVERTAL     *********
405 :
410 FL=1:CH$="":OPEN 15,8,15:OPEN 3,8,3,"@0:MINTA,S,W"
420 FOR N=0 TO 2
430 H1=A(N+1)-A(N):SZ=B(N):GOSUB 1000
440 NEXT N
445 :
450 REM  **********************    IRAS ‾-RE  ****************************
460 :
470 PRINT#3,CH$:CLOSE 3
480 :
500 REM  **********************    OLVASAS ‾-ROL  ************************
510 :
520 FL=0:B(0)=0:B(1)=0:B(2)=0:CH$="":OPEN 3,8,3,"@0:MINTA,S,R":INPUT#3,CH$
526 CLOSE 3:CLOSE 15
530 :
540 REM  ***********   MEZOK ATALAKITASA ES LISTAJA CIKLUSBAN    ***********
550 :
560 FOR N=0 TO 2
570 H0=A(N):H1=A(N+1)-1:GOSUB 1000:B(N)=SZ:PRINT B(N);
580 NEXT N
590 END
1000 REM ***************************************************************
1002 REM * KONVERZIOS RUTIN,  FL=0 ==> 121-ESBOL 10-ESBE                 *
1004 REM *                    FL=1 ==> 10-ESBOL 121-ESBE                 *
1010 REM ***************************************************************
1020 IF FL=1 GOTO 1120
1022 :
1024 REM  ***********************     VISSZAALAKITAS ********************
1030 SZ=0
1040 FOR I=H0 TO H1
1050 FOR J=1 TO 121
1060 IF MID$(CH$,I,1)=MID$(SF$,J,1) GOTO 1090
1070 NEXT J
1080 PRINT"KONVERZIO HIBA":STOP
1090 SZ=SZ+(J-1)*120↑(H1-I)
1100 NEXT I
1110 RETURN
1112 :
1114 REM  ********************    KODOLAS 121-ESBE    *********************
1116 :
1120 FOR I=H1-1 TO 0 STEP -1
1130 H0=120↑I
1140 IF SZ/H0<1 THEN CH$=CH$+"0":GOTO 1170
1150 W=INT(SZ/H0):SZ=SZ-INT(W*H0)
1160 CH$=CH$+MID$(SF$,W+1,1)
1170 NEXT I
1180 RETURN
2000 DATA 1,3,6,9
```

Konverziós tábla sztringben

Az első módszer alapgondolata az, hogy a klaviatúrán 121 olyan jel található, amely sztring változóba téve, a lemezre konverzió nélkül felírható. Ha ezeket a jeleket egy 121-es számrendszer elemeinek tekintjük, az eredeti, 10-es rendszerű számokat programmal átkódolva és a lemezre kiírva, az eredeti helyigénynek törtrészére lesz csak szükségünk. Például ha az ábrázolni kívánt szám 118, ez decimálisan 6 bájtot (előjel, szám, blank, mezőelválasztó) foglal el; a mi számrendszerünkben azonban egyetlen grafikus jel képében, egy bájtban tárolódik.

További sűrítési lehetőség, hogy megvizsgálva a rekordban szereplő mezők helyérték-maximumát, a rekordon belül fix helyre és fix hosszúra deklaráljuk őket, kiírás előtt egyetlen sztringbe láncoljuk (konkatenáljuk) az átalakított változókat, és ezt a sztringet írjuk a lemezre.

Ha egy változó kisebb, mint a számára fenntartott hely, Ø-val töltjük fel az üres helyértékeket. A mezők kezdőpozícióját a rekordon belül egy tömbben tároljuk, és íráskor, olvasáskor ezt használjuk a mező eléréséhez.

A sztring módszer előnye, hogy segítségével megtakarítható a 2 bájtos mezőelválasztó jel és pozitív számokat feltételezve az előjelbájt. A számrendszer konverziós táblájának kialakításakor viszont ügyelnünk kell arra, hogy az ne tartalmazza a , " pi, F1–F7, kurzormozgató, szín, RVS ON/OFF jeleket, mert ezek a sztring vagy a be-kiviteli műveletek közben másképp viselkednek, mint a többi jel. Mi a táblázat kialakításakor Ø-val kezdtünk (ez a 121-es rendszerben is Ø), majd balról jobbra, felülről lefelé haladva, először normál, majd SHIFT, ezután COMM tabuláció mellett ütögettük végig a billentyűzetet.

Az ismertetett *1. program* fő részében a konverziós tábla, a mezők kezdőpozícióit tartalmazó tömb kialakítása és a be-kiviteli műveletek találhatók. Az 1000-es címtől kezdődik a kétirányú konverziót végrehajtó szubrutin. FL=Ø esetében 121-ből binárissá, FL=1 álláskor binárisból 121-esre kódolja a számot.

A bemenő adatok FL = Ø esetén:

CH$ a sztring, amit átkódolunk számmá,
HØ a kódolni kívánt mező kezdőpozíciója CH$-ban,
H1 a kódolni kívánt mező utolsó pozíciója CH$-ban.

Az eredményt az SZ változó tartalmazza.

A bemenő adatok FL = 1 esetén:

SZ az átalakítani kívánt szám,
H1 az eredményül kapott sztring hossza (Ø-val kiegészítés).

CH$-ban a 121-re átkódolt szám található a rutin végén.

Ha a rutint konkatenáció nélkül, csak átalakításra kívánjuk használni,
FL = Ø esetén HØ = 1:
FL = 1 esetén a szubrutin meghívása előtt CH$ = "".

A módszer alkalmazásával egy 30 mezőből álló, decimálisan ábrázolva 4 db egyjegyű, 10 db kétjegyű, 4 db háromjegyű, 5 db négyjegyű és 7 db hatjegyű számot, mely eredetileg 188 bájtban fért el, 60 bájtos rekordba tudtunk elhelyezni, azaz rekordonként mintegy 70 százalékos helytakarékosságot sikerült elérnünk.

Tömb-sztring konverzió

A második módszer annyiban egyezik az előbbivel, hogy a változókat kiírás előtt egy sztringbe gyűjti, és ezt a sztringet írja ki a lemezre. Ebben az esetben azonban a kiindulási alapunk az volt, hogy a tárban az egész számok 2 bájton, a valósak pedig 5 bájton tárolódnak. Az alábbi szubrutin segítségével ezt a tárolási módot ideális megoldásként a lemezen is tudjuk alkalmazni.

A szubrutin működésének lényege, hogy egy adott tömb adatait a tárban levő megjelenési formában bájtonként ASCII karakternek értelmezve, ezekből a karakterekből állít elő egy sztringet. A sztring hossza egész típusú tömb esetén a tömb összes elemszámának kétszerese, valós típusú tömb esetén annak ötszöröse. A hossz természetesen független a tömb egyes elemeinek értékétől. A konverzió visszafelé is igaz, tehát a szubrutin egy sztringet visszair az őt előállító vagy az előállítóval azonos dimenziójú tömbbe. A rutin természetesen csak olyan sztringet értelmez helyesen, amelyet maga állított elő korábban. Mivel egy vektor egydimenziós tömbként is felfogható, a konverzió vektor esetében is érvényes. A tömb dimenziószámára a BASIC szabályai érvényesek. A szubrutin meghívása előtt a következő változóknak kell értéket adni:

VN$ a konvertálandó tömb neve és típusa,
VP$ a sztring neve, olvasás esetén értéke = "",
VD% adatirány, a tömbre vonatkoztatva.

Ezek a változók kötöttek, illetve csak a rutin módosításával változtathatók. A szubrutin hibavizsgálatot végez, és ha hibát talál, azt a VH% nevű változóba adja vissza. VH% = Ø esetén nincs hiba.

A közölt BASIC lista a szubrutint és az ezt kipróbáló főprogramot tartalmazza *(2. program)*. A szubrutin gépi kódú változatát természetesen lényegesen hatékonyabban használhatjuk.

Az adatfeldolgozó rendszerekben általában egy- vagy több dimenziós tömbbe tárolva, tapasztalataink szerint ez a szubrutin igen egyszerű rekordkezelést tesz lehetővé, a lemez tárkapacitását pedig – hardverbeavatkozás nélkül – lényegesen megnöveli. A mi rendszerünkben a 95 mezőből álló, mezőnként átlag négyjegyű számot tartalmazó rekord hagyományos tárolási forma esetén 95 × 8 = 760 bájt hosszú lenne. Az ismertetett tárolással ezzel szemben mindössze 95 × 2 = 190 bájtra van szükség, ami a lemez kapacitásának négyszeres növekedését jelenti.

Végül néhány gyakorlati megjegyzést fűzünk a szubrutin használatához. A konvertálandó tömböt valós vagy egész típusúként deklarálhatjuk a legelső DIM utasításban, ezáltal a tömb nevét kereső FOR ciklus rövidebb. Tömbből való olvasás előtt a VP$ sztringváltozó kezdeti értéke VP$ = "". A szubrutin maximálisan 127 elemű egész típusú tömböt és 51 elemű valós típusú tömböt képes feldolgozni.

GOSZTOLA FERENC–VÖLGYI CSIK JÓZSEF

2. program

```
100 REM FOPROGRAM
102 DIM MZ%(4,3,2),HZ%(4,3,2)
104 FOR I=0TO4:FOR J=0TO3:FOR K=0TO2:MZ%(I,J,K)=5:NEXT K:NEXTJ:NEXTI
106 FOR I=0TO4:FOR J=0TO3:FOR K=0TO2:HZ%(I,J,K)=0:NEXT K:NEXTJ:NEXTI
108 VP$="":VN$="MZ%":VD%=2:GOSUB 9000
110
112 REM IRAS LEMEZRE VAGY KAZETTARA  VP$-BOL
114
116 REM OLVASAS LEMEZROL VAGY KAZETTAROL  VP$-BA
118 VN$="HZ%":VD%=1:GOSUB 9000
120
122 REM FELDOLGOZAS
124
126 END   : REM FOPROGRAM VEGE
128
130 REM ************************************************************************
132
9000 REM TOMB-STRING KONVERZIO
9002 REM HASZNALT VALTOZOK          *** VN$="X%"  VALOS TIP. TOMB NEVE ***
9004 REM                            *** VN$="X%" EGESZ TIP. TOMB NEVE ***
9006 REM                            *** VP$ STRING NEVE               ***
9008 REM                            *** VD% ADATIRANY  1=IR  2=OLVAS  ***
9010 REM                            *** VH% HIBAKOD                   ***
9012 REM                            *** V1,V2 TOMBOK KEZDO- ES VEGCIME ***
9014 REM                            *** VE,VV ADATOK KEZDO- ES VEGCIME ***
9016 REM                            *** VH%=1 TOMBNEV NEM DEFINIALT   ***
9018 REM                            *** VH%=2 A TOMB TUL NAGY         ***
9020 REM                            *** VH%=3 ADATIRANY HIBA          ***
9022
9024 V1=0:V2=0:V3=0:V1=0:V3$="":VE=0:VV=0
9026 VH%=0:V1=ASC(LEFT$(VN$,1)):V2=ASC(MID$(VN$,2,1))
9028 IF LEN(VN$)=2 THEN V1=V1+128:V2=V2+128
9030 FOR VJ=PEEK(47)+PEEK(48)*256 TO PEEK(49)+PEEK(50)*256
9032 IF PEEK(VJ)=V1 AND PEEK(VJ+1)=V2 THEN 9036
9034 NEXT VJ:VH%=1:RETURN
9036 IF PEEK(VJ+3)>0 THEN VH%=2:RETURN
9038 VE=(PEEK(VJ+4)+2)+VJ+5 :VV=VJ+PEEK(VJ+2)-1
9040 IF VD%=1 THEN 9046
9042 IF VD%<>2THEN VH%=3:RETURN
9044 VP$="" FOR VI=VE TO VV:VP$=VP$+CHR$(PEEK(VI)):NEXT VI:RETURN
9046 FOR VI=VE TO VV:V3$=LEFT$((MID$(VP$,VI-VE+1,1)+CHR$(0)),1):POKEVI,ASC(V3$)
9048 NEXT VI: RETURN
9050 REM SIR VEGE
9052
9054 REM ************************************************************************
```

Építsünk számítógépet! VIII.

Ebben a részben eredetileg a központiegységkártya ismertetését akartam elkezdeni, de engedve a nagyszámú olvasólevélben szereplő kívánságnak, a gép teljesítőképességével foglalkozom. A minimális, egy közepes, egy teljes kiépítésű változatot, valamint amerikai cégek által az SS–5Ø buszhoz (ami a mi gépünkben is van) elkészített fejlesztéseket foglalom össze.

A *minimális változat* az alapkártyából; az 1 k RAM-ot tartalmazó videokártyából; a 68Ø3-ban levő 128 bájt RAM-on kívül más RAM-ot nem tartalmazó, egy 2716-ban tárolt ROM monitoron kívül más ROM-ot nem tartalmazó központiegység-kártyából; a billentyűzetillesztőből; egy tévévevőből és egy közönséges magnetofonból áll.

Ennek a változatnak 64 × 16 karaktert megjelenítő képernyője, 11Ø–24ØØ Baud átviteli sebességű magnetofon-háttértára (jó minőségű magnetofon és kazetta esetén 48ØØ Baud) van. A 2 k ROM és 128 bájt RAM (+1 k a képernyőnek) csak kis tárigényű gépi kódú programok tárolására elég.

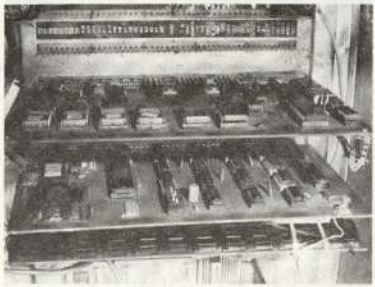
Elégséges a rendszert kiszolgáló SIMON COS (kazettás operációs rendszer) tárolására, melynek parancsai:

- két változatú programfuttatás,
- BASIC hideg indítás,
- memóriavizsgálat, -módosítás,
- karaktersorozat (48 karakterig) kikeresése,
- regisztervizsgálat,
- automatikus programáthelyező, adatáthelyező, címmódosító parancs egyidejűleg legfeljebb 8 tömb módosítására, írásvédelem figyelembevételével,
- tárterület-törlés,
- töréspont be/kiiktatás,
- csatornakiválasztás,
- tárkiíratás,
- adatbeolvasás magnetofonról, beállítható sebességgel,
- két tárterület tartalmának összehasonlítása,
- adatkivitel magnetofonra, beállítható sebességgel,
- időkésleltetés kiadása,
- a 68Ø3 belső RAM-területének be/kikapcsolása,
- videovezérlések (kezdeti feltételek beállítása, fénypont fel-le, jobbra-balra, sor/lap elejére, törlés),
- megszakításkezelés.

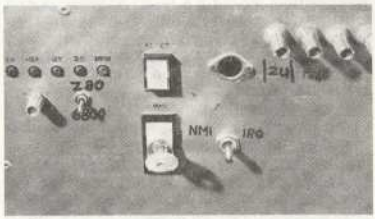
Minden parancs hívható szubrutinként is. Ez a változat alkalmas a rendszer működését vizsgáló programok tárolására is.

Az *átlagos kiépítésű változat* ugyanezeket a kártyákat tartalmazza, de a videokártyán 6 k RAM, a központiegység-kártyán 16 k ROM és 16 k RAM (a kártya módosításával 32 k ROM és 64 k RAM is megvalósítható) van.

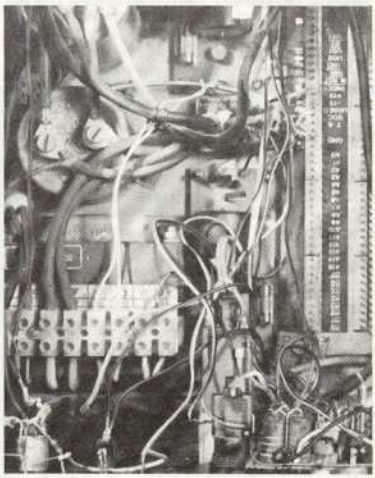
Ennél a változatnál a képernyő grafikus is lehet 256 × 192 pont felbontással, és monitor használata esetén 8Ø × 24 karakterrel.



1. ábra



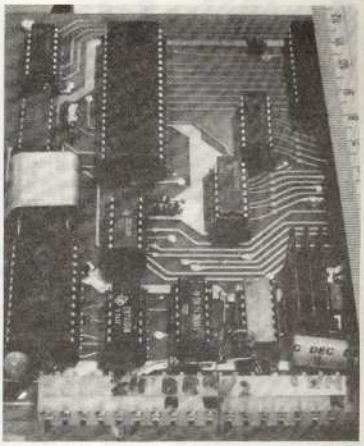
2. ábra



3. ábra

A géphez 4, 8, 11 k helyfoglalású BASIC-fordítók állnak rendelkezésre. Programfejlesztéshez assembler, disassembler, ún. debugger (ez az előző kettőn felül például intelligens szimulátort, töréspontokkal számos paramétert, vizsgálót is tartalmaz) van.

Teljes kiépítés. A gép bővíthető 16 k RAM befogadására tervezett, 16 csatornás A/D átalakító-, soros illesztő-, párhuzamos illesztő-, magnetofonkezelő-, hajlékonytárcsás lemezegység kezelő kártyákkal. A központiegységkártya lehet 68ØØ, 68Ø9, Z8Ø mikroprocesszorú is. Ezek közül bármelyik kettő egyidejűleg is benne lehet a gépben, mint ahogy az 1977-es gépünkről készült, *1. képen* látható.



4. ábra

A két központiegység-kártya közül az aktívat (a másik letiltása mellett) egy egyszerű kapcsoló választja ki, ahogy az ugyanannak a gépnek két oldaláról készített *2.* és *3. képen* látható. Ez utóbbin követhető, hogy a két kivezetéses kapcsoló egyik vezetéke az NMI kapcsolóhoz, a másik a busz UD1-vonalához vezet.

Az átkapcsolás a mikroprocesszort a buszról lekapcsoló lábra adott jellel történik *(4. kép)*. Mi ezt az egyszerű megoldást – amelyet utánunk mintegy másfél évvel kezdett el gyártani a Microsoft cég az Apple II típusú géphez, SOFTCARD néven, és amelynek különböző változataiból már millión felüli darabszám készült – szabadalmaztatni kívántuk. *Megvalósíthatatlanság* indokával azonban kérelmünket elutasították.

Az SS–5Ø buszhoz kapható egyéb fejlesztések: a 68ØØ, 68Ø9, 68Ø3 kártyáink lehetővé teszik a FLEX és az OS9 operációs rendszerek használatát, a Z8Ø kártya pedig, monitorával együtt a CP/M 2.2 változatát képes használni. Ezeken keresztül gyakorlatilag mindenféle szoftver elérhető.

A hardverlehetőségek között megemlítem a 256 k DRAM, a mindenféle lemezegységillesztő, a nagy felbontású (1Ø24 × 512 képpontú) grafika és a folyamatmérő/vezérlő kártyákat, és végül, de talán ez a legfontosabb, a 68ØØ8 és 68ØØØ típusú mikroprocesszoros kártyákat.

DR. SIMONYI ENDRE

Mi már építünk!

Mint már beszámoltam róla, elkészült épülő számítógépem billentyűzete. De első próbálkozásra ritkán sikerülnek jól a dolgok, és sajnos ebben az esetben is ez történt. Legelőször csak néhány forrasztási hibára derült fény, amelyeket hamar kijavítottam. Nem ez okozta a fő gondot. A gyakorlat ugyanis azt mutatta, hogy bármelyik billentyűt nyomom le, mind a tizenhárom vezetéken folyik áram.

Sokáig kerestem a hibát. Először arra gondoltam, hogy a vezetékek érintkeznek, de máshol volt a baj: a földelésnél. Kiderült, hogy amikor az egyes vezetékeket leföldeltem, nem akadályoztam meg az áram átfolyását a többi vezetékre. Néhány dióda alkalmazásával megszüntettem a hibát. Előbb ugyan egy másik NYÁK-ot kellett készíteni, de ez már nem okozott különösebb nehézséget.

A kész billentyűzetet a *képen* láthatják. Az érintkezők – vasútmodellezéssel foglalkozó olvasók biztosan felismerik – az ún. váltóirányító pult kapcsolói. Az egyes billentyűk kódolására diódamátrixot alkalmaztam.

Az illesztőt nem számítva, mindössze 700 forintba került a billentyűzet, amiben szerepe van annak, hogy a hat érintkezőt tartalmazó kapcsolópult ára csak 40 forint volt.

Végül az illesztőrészről néhány szót. Használatát az indokolja, hogy a számítógépnek csak 8 adatvezetéke van arra a célra, hogy kommunikáljanak vele, míg mi a karaktereket 13 biten kódoltuk. A billentyűzetillesztő tehát azt a célt

szolgálja, hogy a mikroprocesszornak alkalmas időben, számára feldolgozható információt küldjön.

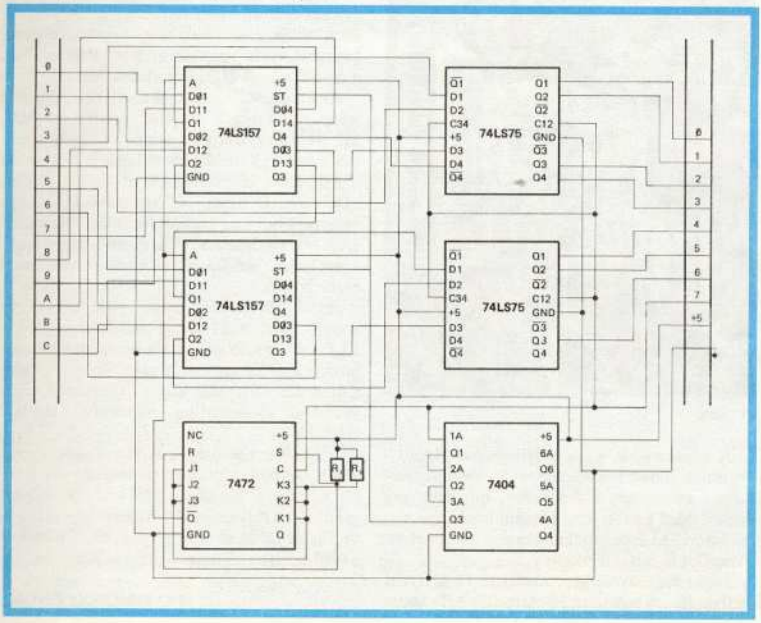
Ez a következőképpen történik. A kódot egy 6 és egy 7 bitből álló részre választja szét a rendszer, és először a kód első 7 bitjét küldi ki, majd pedig a második 6 bitet. A szétválasztást a 74LS157-ek végzik, a 74LS75-ök pedig a felhasználásig tárolják az információt. A rendszer vezérlését a 7472 végzi, a 7404 csak késleltetésre van.

Mivel legfeljebb 7 adatvezetéket használunk, a 8. vezetéken küldhetjük ki az engedélyező jelet.

Gondolom, sokaknak feltűnt, hogy a billentyűzetillesztő elvi kapcsolási rajza hasonlít a dr. Simonyi Endrének a lap 1984/2. számában közölt cikkében szereplő rajzhoz, de sokban eltér tőle. A magyarázat az, hogy az itt közölt rajz nemcsak a rendszer vezérlését tünteti fel, hanem a célnak is jobban megfelel, mivel ennél a megoldásnál egyszerűbb szoftverrel lehet a kezelést végezni.

GATTYÁN ANDRÁS

A billentyűzetillesztő elvi kapcsolási rajza



Kazetta-szabvány

A mikrogépek gyártásának már kezdetén látható volt, hogy a termékek csereszabatossága sokkal kevésbé lesz biztosított, mint a nagyobb gépeknél. Ott sokkal kevesebb cég volt a piacon, és volt egy, az IBM, amely a hardver- és szoftverválasztékot döntő módon meghatározta. Itt azonban sok kis cég versenyzett, eltérő hardver- és szoftverelképzelésekkel, eszközökkel.

A hardverpiacon igazán sikeres szabványosítási kísérlet nem volt, mert mire megszületett az egyébként is laza, S–1ØØ buszszabvány, már a mikrogépek mintegy 80 százalékát a Tandy, a Commodore és az Apple cégek gyártották, amelyeknek egyike sem fogadta ezt el.

Sokkal eredményesebbek voltak a szoftvertermékek egységesítésére irányuló kísérletek. A lemezoperációs rendszereknél a Digital Research CP/M-je ugyan sok változatban készült el, és a „három nagy" egyike sem használta, de valamennyi, tömeggyártásba került géphez elkészült a szoftver- és hardverillesztés. Ez utóbbi a 65Ø2 típusú mikroprocesszorú gépeknél Z8Ø-as központiegység-kártya beépítésével és az aktív központi egység kapcsoló kiválaszthatóságával járt együtt. Az első ilyen, kereskedelmi forgalomba került kártya a Microsoft cég SOFTCARD nevű kártyája volt.

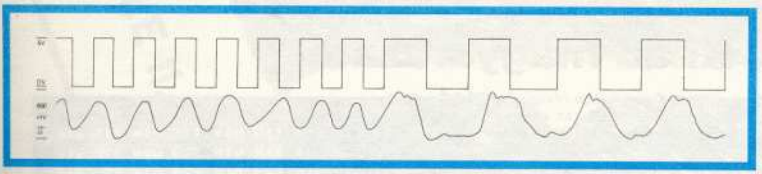
A BASIC-fordítók között a Microsoft cég róla elnevezett fordítóinak különböző változatait gyártották a „nagyok" is. Ezek azonban sok közös tulajdonságuk ellenére egymás programjait nem értik. Még az azonos elnevezésű és jelentésű utasításokat, függvényeket sem, mivel ezekhez különböző kulcsszókódok tartoznak.

Ez a félig sikerült szoftveregységesítés is csak a 80-as évek elejére jött létre, de a zűrzavar már kezdettől fogva megvolt. Ennek csökkentésére a Byte című szaklap összehívott egy konferenciát, amelyen a gyártók (akkor még a „három nagy" egyike sem gyártott mikrogépet) és a főbb felhasználók megkísérelték az álláspontok közelítését. Az egyetlen, amiben megállapodtak, a kazettajel volt. Az erre vonatkozó szabvány, a konferencia színhelyének nevéről, mint Kansas City-szabvány vált ismertté, és bár eleinte elterjedt, a lemezrendszerek elterjedésével fokozatosan visszaszorult. Mai jelentősége abban van, hogy használata lehetővé teszi a programok, adatok átvitelét gépről gépre.

Mit tartalmaz ez a szabvány? Előírja, hogy

- a jel négyszög alakú periodikus, tehát két feszültségszintű, azaz bináris,
- az információhordozó egy megadott idő alatti periódusszám, konkrétan 1/3ØØ s alatt négy (ez a Ø) vagy nyolc (ez az 1) periódus *(l. ábra)*,
- a megengedett frekvenciaingadozás ± 3Ø%, vagyis a magnetofon sebességingadozására gyakorlatilag érzéketlen,
- az amplitúdóingadozás érdektelen,
- a jelformatorzítás is érdektelen. Ez utóbbi kettő oka az, hogy az igazi információ csak a feszültségváltozás irányváltozásainak száma. Így például az *ábrán* látható torzulása a jelsorozatnak 1 : 5 amplitúdóingadozás mellett sem okozott problémát egy korábbi gépünknél.

A szabvány nagy előnye a könnyű megvaló-



```
5 REM1.PROGRAM:KC-TOELTOE
10 CLS:CLEAR500,14335:A=14346:PR
INT"BEHELYEZENDOE A KAZETTA! A M
AGNETOFONT OLVASAASRA KELL AALIT
ANI!":B$="HA KEESZ ADJON BE EGY
SZAAMOT!":PRINTB$:INPUTU:POKE&H3
803,0:POKE&H3802,57
20 FORI=0TO107:READA$:POKEA+I,VA
L("&H"+A$):NEXTI:MOTORON:EXEC&H3
80A:E=256*PEEK(&H3802)+PEEK(&H38
03):PRINT"VEEGCIM:";E
30 E=256*PEEK(&H3802)+PEEK(&H380
3)
40 PRINT"A BEOLVASAAS KEESZ! CSE
REELJEN KAZETTAAT!":PRINTB$:INPU
TU:INPUT"PROGRAMNEEV";C$:CSAVEM
C$,&H4C00,E,&H4C00:STOP
60 DATABE,38,02,8D,0D,A7,80,CE,0
4,00,A7,C4
70 DATABF,38,02,20,F2,39,36
170 DATA04,8D,2D,2D,FC,8D,29,8D,
27,2D,F6,C6,08,F7,38,00
180 DATA4F,8D,11,44,8D,30,2E,02,
8A,80,8D,08,7A,38,00,2E
190 DATAF2,37,04,39,1F,12,8E,01,
04,30,82,26,FC,1F,21,39
200 DATA36,02,4F,4C,F6,FF,20,C4,
01,F1,38,01,27,F5,F7,38
210 DATA01,80,0E,37,02,39,F6,FF,
20,C4,01,F7,38,01,8D,E0
220 DATA8D,DE,2E,04,8D,DA,8D,D8,
39
```

1. lista

síthatóság olcsó eszközökkel, olcsó illesztő, magnó, szalag. Hátránya a lassúsága: átviteli sebessége mindössze 300 Baud. A kis sebesség és torzításérzéketlenség lehetővé tette a programok továbbítását telefonvonalakon is magnetofonhoz és vissza.

Az eddigiek alapján a KC-illesztő hardver és szoftver egyszerűen megvalósítható és jól használható eszköz. Az íróhardvernek képesnek kell lennie TTL-szintű, majd ezekből a magnó által venni képes jelek létrehozására. Ehhez egy párhuzamos csatorna egy bitje, egy 7407, néhány ellenállás és kondenzátor elég. Az olvasónak a magnó által torzított jelet kell TTL-jellé alakítania, majd azt a mikroprocesszorhoz továbbítania. Ehhez az előző elemek azzal az eltéréssel elégségesek, hogy a 7407 helyett egy összehasonlító elem (például LM 339) szerepel.

Az olvasószoftver a kapott logikai 0,1 sorozatokból a küldött bitsorozatot kell, hogy azonosítsa és tárolja. Ennél többet egy univerzális olvasóprogramtól nem várhatunk, mert még azt sem sikerült szabványosítani, hogy mit jelentsenek a bitek. A legelterjedtebb változatnál ún. 8 bites adatot, start-, stopbiteket (esetleg párosságellenőrző bitet) visznek át. A startbit (logikai 1) jelzi az adat elejét, melyet követ az adat, végül a stopbit (logikai 0). Az adatbájt általában hexadecimális (tizenhatos számrendszerbeli) számként átvitt ASCII-kód. Ez azt jelenti, hogy a bájtot két 4 bites (azaz hexadecimális) számra bontjuk, majd ASCII-karakterré alakítjuk. Ez utóbbinál, ha a hexadecimális számjegy 0–9, akkor hozzáadunk 48-at, ha pedig A–F, akkor 65-öt. Az olvasónál természetesen a folyamat fordított.

Ilyen típusú adatok beolvasására szolgáló BASIC program látható az *1. listában*. A programot DRAGON típusú gépre írtam. A DRAGON-BASIC egy kiterjesztett, színes Microsoft-BASIC, amelynek különleges tulajdonságait alig használtam ki, azért, hogy a program más Microsoft-BASIC-kel rendelkező gépeken (például HT–1080Z, Commodore gépek, Apple II) csaknem változatlanul futtatható legyen.

A módosításokhoz: a CLEAR utasítás első száma a karaktersorozatok részére foglal helyet, míg a második a BASIC programterület felső határát adja meg; a §H-jel hexadecimális számot jelöl; az EXEC a gépi kódú program behívása; a CSAVEM utasítás gépi kódú programot visz kazettára névvel, kezdő- és végcímmel, valamint a program indító címével.

A gépi kódú rész természetesen mikroprocesszor-függő. Így az csak 6809 típusnál használható változatlanul. Az átírás a *2. listához* fűzött megjegyzésekkel azonban egyszerű. A program a 3900H címtől tölt, függetlenül attól, hogy a beolvasandók milyen címeken futnak. A DRAGON-nál a beolvasható mennyiség kb. 25 k lehet (a továbbfeldolgozás miatt azonban csak ennek a fele).

A program gépi kódú része a DRAGON-képernyő bal felső sarkát „ablaknak" használja (Commodore 64-nél ez szintén a 0400H címre töltéssel történik), és a töltés végén állandósuló jel (ez a később említett formátumnál "9") megjelenésekor kell a magnót leállítani. Ezután RUN 30 paranccsal a beolvasottakat TRS–80 formátumban, amit a DRAGON is használ, kazettára viszi. Ehelyett géptől, kiépítéstől függően, saját kazettára, illetve lemezre kivitelt kell itt megvalósítani.

Az így kivitt program visszatöltve általában nem futtatható, még akkor sem, ha a mienkkel azonos típusú gépen készült, és a mi gépünkön is megszokott formátumban. A szalagra vitt adatokat ugyanis általában sorokra-blokkokra bontva viszik ki, melyek mindegyikét azonosító-val-típusjellel (esetleg kezdőcímmel is), valamint ellenőrzőszámmal egészítik ki. Ezeket értelmezni kell, és figyelembevételükkel lehet csak a programot úgy elhelyezni a memóriában, hogy az futtatható legyen.

Ennek megvalósítására szolgáló BASIC program látható a *3. listában*. Ez a program a Motorola cég ún. S1–S9 formátumát dolgozza fel. A program a kívánt kezdő- és az előző program által kiírt végcímmel dolgozik. Amennyiben ez nem lehetséges, úgy ezt jelzi, és újat kér. A sorok/blokkok kezdőcímeit kiírja, így ha azok nem egyenletesen követik egymást, valami probléma adódott. A hibás részt RUN 200 paranccsal vizsgálhatjuk. A program csak 2800H cím feletti területre tölthet! Ez alatti területre csak úgy tölthetünk, ha először a megengedett területre töltünk, majd utána külön áthelyezzük.

Hogyan dolgozik a program? Először is kere-

```
Címek: (Kezdőcím helye): 3802-3H
       REF: 3801
       Bl(tszám): 3800
       PIA-cím: FF20
       A(blakcím): 0400

A program:

BE 3802   S LDX      K     Töltsd a tárolócímet!
8D 0D       BSR      T     Olvass be egy adatot!
A7 80       STA      ,X+   Tárold a megadott címre!
CE 0400     LDU      #A    Tedd az „ablakba"!
A7 C4       STA      ,U
BF 3802     STX      K     Tárold a tárolócímet!
20 F2       BRA      S     Újra az elejére
39          RTS
36 04     T PSHU     B     Védd meg a B-regisztert!
8D 2D     T1 BSR     V     Kapod a startbitet
2D FC       BLT      T1
8D 29       BSR      V     Jön a bitsorozat...
8D 27       BSR      V
2D F6       BLT      T1    Ha hibás, vissza
C6 08       LDB      #8    Állítsd be a bitszámot!
F7 3800     STB      BI
4F          CLR      A     Töröld az adattárolót!
8D 11       BSR      I     Várj egy bitnyi időt!
44        T2 LSR     A     Léptess a következő bithez!
8D 30       BSR      AA    Bitállapot?
2E 02       BGT      T3    Ha „0", folytasd!
8A 80       ORA      #$80  Ha „1", változtass!
8D 08     T3 BSR     I     Várj egy bitnyi időt!
7A 3800     DEC      BI    Ha van még bit...
2E F2       BGT      T2    folytasd!
37 04       PULU     B     Töltsd vissza B-t!
39          RTS            Térj vissza!
1F 12     I TFR      X, Y  Védd meg az X-regisztert!
8E 0104     LDX      #$0104  Baud-sebesség állandó*
30 82     I1 LEAX    ,X    Visszaszámlálás
26 FC       BNE      *-1
1F 21       TRF      Y,X   Töltsd vissza az X-regisztert!
39          RTS            Térj vissza!
36 02     V PSHU     A     Védd meg az A-regisztert!
4F          CLR      A     Töröld a számlálót!
F6 FF20     LDB      PIA   Olvass be a magnóról!
C4 01       ANDB     #1
F1 3801     CMPB     REF   Ha azonos a referenciával...
27 F5       BEQ      V1    várj a változásra!
F7 3801     STB      REF   Tárold az új értéket!
80 0E       SUBA     #$0E  Frekvencia állandó*
37 02       PULU     A     Töltsd vissza az A-regisztert!
39          RTS            Térj vissza!
F6 FF20   AA LDB     PIA   Olvass be a magnóról...
C4 01       ANDB     #1    a referencia tárolóba!
F7 3801     STB      REF
8D E0       BSR      V     Határozd meg a hullámformát!
8D DE       BSR      V
2E 04       BGT      AA1
8D DA       BSR      V     „1"-hez tartozó idő
8D D8       BSR      V
39        AA1 RTS          Térj vissza!

* Ezek az értékek 894 kHz frekvencián járatott 6809-re vonatkoznak. Más esetekre az egyes utasítások végrehajtási ideje alapján kell a megfelelő értéket meghatározni.
```

2. lista

si a bevezető ASCII 00 kódsorozatot (legalább 5), majd az "S1" blokk-kezdést. A blokk kezdőcíme alapján megvizsgálja, hogy a megkívánt végleges címterület nem fedi-e át a betöltési címterületet. Ha igen, módosíthatjuk a megkívánt címet.

Átdolgozása más formátumokra a következők segítségével történhet. A bevezető, blokkfejrész-vizsgálat mindig szükséges. Eltérés csak a karakterek kódjában, számában lehet (5ø, 7ø, 8ø, 13ø-as sor). A tetszés szerinti címre tölthető formátumnál természetesen a blokkból címki-

```
10 REMS1-S9 MOTOROLA FORMAATUM F
ELDOLGOZOD
20 CLS:CLEAR500,&H27FF:INPUT"PRO
GRAMNEEV, KIVAANT KEZDOECIM, ERE
DETI VEEGCIM";C$,A,E:CLOADM C$,A
-19456:C=0:E=E+19456-A
30 FORI=A TO E:B=PEEK(I):IFB<>0T
HEN50
40 C=I:I=E
50 NEXTI:IFC<>0THEN70
60 PRINT"NINCS ELEJE!":STOP
70 A1=0:FORI=C TO E:B=PEEK(I):IF
B<>0THEN100
80 FORJ=0TO4:B=B+PEEK(J+I):NEXTJ
:IFB<>0THEN100
90 C=I:I=E
100 NEXTI:IFB<>0THEN60
105 B1=0
110 D=0:IFC<E THEN130
120 PRINT"VEEGE!":IFB1=0THENPRIN
T" CSAK 0-SOROZAT!"
125 STOP
130 FORI=C TO E:B=PEEK(I):A1=PEE
K(I+1):A1=100*B+A1:IFA1<>5331THE
N150
140 C=I+4:I=E:D=1:A$=CHR$(PEEK(I
+2))+CHR$(PEEK(I+3)):A1=VAL("&H"
+A$):B1=B1+1:IFB1<>1THEN150
145 C1=C-4
150 NEXTI:IFD=0THEN120
160 A$="":FORI=0TO3:A$=A$+CHR$(P
EEK(I+C)):NEXTI:B=VAL("&H"+A$):I
FB<A THEN185
170 IFB>E THEN185
180 PRINT"A TALAALT CIM:";B;" BE
LEESIK A MEGADOTT TARTOMAANYBA "
;A;"-";E;". A TALAALT KEZDOECIM:
";C2:GOTO20
185 IFB1<>1THEN190
186 C2=B:PRINT"A TALAALT KEZDOEC
IM ";B;". MOODOSITJA(I/N)?":INPU
TU$:IFU$="N" THEN190
187 INPUT"UJ CIM";B2:IFB2>&H2800
THEN189
188 PRINT"NEM MEGENGEDETT!":GOTO
186
189 B2=B2-C2:C2=C2+B2
190 B=B+B2:PRINTB;" ";:FORI=0TO
2*A1-1:A$=CHR$(PEEK(C+I))+CHR$(P
EEK(C+I+1)):POKEB,VAL("&H"+A$):C
=C+2:B=B+1:NEXTI:C=C+6:GOTO110
200 INPUT"A HIBA KEZDOE EES VEEG
CIME";X,Y:D1=2*(4+A1)*(X-C2)/A1:
D2=2*(4+A1)*(Y-C2)/A1:FORI=D1 TO
 D2:PRINTHEX$(PEEK(I+C1));:NEXTI
:STOP
```

3. lista

olvasás és címvizsgálat (160, 180, 186-os sor) elmarad. A programban használt kevéssé ismert jelek: a CLOADM utasítás adott nevű gépi kódú programot tölt be a megadott címtől; a HEXSI függvény decimális egész számot alakít át hexadecimálissá.

Az „ablak" használata megkönnyíti ismeretlen formátumok feldolgozását is, hiszen az ismétlődések a „szemünk előtt" játszódnak le. Amennyiben nem adódik szabályosság, és csak egy-egy „értelmes" karaktert látunk, úgy lehet, hogy nem egy stopbites a formátum. Ezen a gépi kódú program megfelelő részének módosításával segíthetünk.

A második program indítása előtt DRAGON 64-nél beadandó: EXEC 48000.

DR. SIMONYI ENDRE

Ki ad magyarázatot?

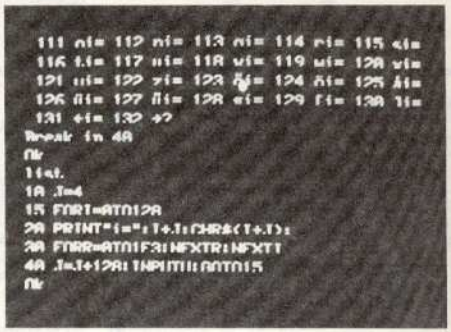
Ebben a sorozatban olyan „furcsaságokat" közlünk (az amerikai szakirodalomban BUG), amiket egy-egy gép elkövet, és amikre a géphez adott dokumentációkban nincs magyarázat.

Reméljük, hogy akadnak olyan olvasók, akik képesek magyarázattal szolgálni, és ezt el is küldik rèszünkre. Ezeket, ha valóban magyarázatok, közöljük.

Magyarázatot a gyártó is adhat! Várjuk a „furcsaságokat"!

Elsőként lapunk 1984/6. számában, a Primóval kapcsolatban megjelentekre várjuk a válaszokat. Kiegészítésül az eredeti cikk 3. fényképén látható eredményt szolgáltató programot is közöljük.

DR. SIMONYI ENDRE



Közérdekű válaszok az olvasóknak

1. A rovatszerkesztő az időveszteség és a levelek továbbküldözgetésének csökkentése céljából felkéri az olvasókat, hogy a cikkekben nem említett témákkal kapcsolatban keressék meg

- elsősorban a hozzájuk legközelebb eső helyen működő klubot (ezekről tájékoztatást kaphatnak a Neumann János Számítógéptudományi Társaságban),
- ha ez eredménytelen lenne, úgy a HCC budapesti szervezetének megfelelő szekcióját (tájékoztatás ugyanott),
- és csak legvégső esetben a rovatszerkesztőt.

2. Azoknál a cikkeknél, ahol az szerepel, hogy „A Klub kellő számú érdeklődő esetén vállalja" valaminek az elkészítését, vegyék a szöveget komolyan, és ne küldjenek feleslegesen sürgető leveleket. Ettől a feltételek nem változnak meg!

3. Könyvek, szaklapok, adatlapok, kapcsolások, részegységek, alkatrészek beszerzése ügyében tanácsom azonos az 1. pontban leírtakkal. Ilyeneket nem tudok küldeni!

4. Akik a HCC tagjai akarnak lenni, írják meg az NJSZT-nek, hogy melyik szekcióba akarnak belépni, és kérjék, hogy levelüket továbbítsák a szekciónak.

5. Csak olvasható névvel és címmel ellátott levelekre tudok válaszolni, ezért kérem, hogy ezeket az adatokat vagy írógéppel vagy nyomtatott nagybetűkkel írják!

6. „Építsünk számítógépet!"

A gép építéséhez szükséges ismeretek. Mindenfajta számítógép megépítéséhez (és ez alól ez a gép sem kivétel) elengedhetetlenül szükséges kellő forrasztási, szerelési gyakorlat. Ennek hiányában csak drága alkatrészek tönkretétele árán tanuljuk meg ezeket a műveleteket. Természetesen ezenfelül bizonyos tájékozottságra is szükség van az elektronikus áramkörök és a gépbe beépítésre kerülő alkatrészek terén. Ezeket az ismereteket terjedelmigényük miatt ebben a sorozatban nem adhatjuk. (A Microcomputing című szaklapban egy olyan sorozat, amely ezeket az alapismereteket is nyújtotta, és a végén egy nagyon egyszerű mikroszámítógépet, 24 részből állt, és egy-egy rész terjedelme a mienkének háromszorosa volt.)

A gép építéséhez szükséges eszközök. Feltételezve, hogy az áramköri kártyák készíttetése sikerül, a gépépítőknek forrasztás, ellenőrző mérések végzése, összeszerelés lesz a feladatuk. (Az élesztést ez a felsorolás nem tartalmazza!) Ehhez jó forrasztópáka, statikus és dinamikus (de legalább statikus feszültség, áram, ellenállás), elektronikus jellemzők méréséhez szükséges eszközök nélkülözhetetlenek. Az irógéppillesztő eredeti változatának építőire mechanikai megmunkálás is vár, így ilyen irányú ismeretekre és eszközökre is szükség van.

A szerző lehetőségeinek korlátai. Kénytelen vagyok ismételten leírni, hogy nem áll mögöttem egy világcég, amely esetleg reklám céljából hajlandó lenne az olvasóknak számítógépet, áramköri kártyát, alkatrészeket, filmeket, rajzokat küldeni – ingyen. Ez utóbbiakat és a beültetés nélküli kártyák fényképeit, részletes elvi rajzát, működési leírását a sorozatban ismertettem, illetve folyamatosan ismertetem. Az ezekkel kapcsolatos levelekre ezért külön nem válaszolok.

A gép teljesítőképessége. Eredeti elképzelésem az volt, hogy az ezzel kapcsolatos rész az egyes egységek és összeállításmódjuk ismertetése után, a sorozat végén következik majd. A sok kérdés miatt azonban a sorozat mostani száma erről szól.

Az olvasók tapasztalatai. Ha közérdekűek, közöljük azokat. Így került sor a módosított billentyűzetilleszto közlésére.

Közreműködés a „gyártásban". Több olvasó is ajánlkozott arra, hogy díjazás fejében részt venne a gépépítésben, bemérésben, élesztésben. A konkrét ajánlatokat és kéréseket, a segítséggel kapcsolatos tapasztalatokat készek vagyunk közvetíteni és hozzájárulás esetén közzétenni.

A szerző elérésmódja. Minden, a μM-hoz érkező és nekem címzett levelet, telefonüzenetet megkapok, ezért kérem, hogy a továbbiakban is kizárólag a szerkesztőség címén keressenek meg.

Megrendelés. Sokan szeretnék megrendelni az áramköri kártyákat. Az egyik olvasó javasolta, hogy írjanak azok, akik az ár ismeretében megrendelnék. Tehát: írjanak, akik a 3500 forintot meg nem haladó áron megrendelnék az alapkártyából, központiegység-kártyából, képernyőhajtó kártyából álló készletet!

DR. SIMONYI ENDRE

- Hőcserélő méretezés
- Kémény méretezés
- Klíma szimulációs program
- Épületek hőszükséglet számítása
- Építészeti rajzolóprogram
- Mesterséges szellőzőberendezések méretezése
- Kétcsöves szivattyús központi rendszer méretezése
- Deflektoros kéményméretezés

A programok
más típusú számítógépekre is átdolgozhatók.

APPLE II. és a vele kompatíbilis számítógépekre ajánljuk programjainkat

A programok megrendelhetők az alábbi címen:

Alkotó Ifjúság Egyesülés
Számítástechnikai Iroda
1519 Budapest, Pf.: 330.
Telefon: 112-666, 113-608

A technológiák szerepe

A mikroelektronika oly mértékben a mikroszámítógépes technika megalapozója, hogy túlzottan divattá vált szerepének egyoldalú hangsúlyozása, ami néha már frázisszerű pufogtatásnak hat. Ennek az a veszélye, hogy a mikroelektronika afféle univerzális „bölcsek kövének" tetszik, ami minden bajra önmagában megadja a megfelelő megoldást. Másként mutatkozik azonban, ha azoknak a mélyebb összefüggéseknek az oldaláról nézzük, amelyeken keresztül folyamatosan hatást gyakorolt és gyakorol a modern informatika arculatának kialakulására. És egész másképpen jelentkezik akkor, ha a társtechnológiákkal együtt szemléljük befolyását, és nem utolsósorban az emberi tényezőt sem felejtjük el.

Sok-sok MOS memória

Az ún. fémoxid-félvezető (Metal–Oxide–Semiconductor – MOS) típusú memóriaáramkörök kifejlesztése előtt félvezető-technológiával csak bipoláris flip-flop komponenseken alapuló áramköröket gyártottak a félvezetőgyártók. Az utóbbiak nem tudták kiszorítani az operatív tárak felépítéséhez addigra széles körben elterjedt ferritgyűrűs megoldást, és elsősorban a processzoron belüli tárelemek (regiszterek) kialakításában kerültek alkalmazásra. A nagyobb elemsűrűség és kisebb áramfelvétel miatt a MOS technológia viszont lehetővé tette az áramkörök integráltsági fokának oly mértékű növelését, aminek eredményeképpen a félvezető memória egyre versenyképesebb lett a nehézkesen tekercselhető, drága és nagy térfogatigényű ferritgyűrűs megoldással szemben.

Ezt a lehetőséget ismerték fel az 1968-ban alapított Intel cég vezetői, és felismerésüket hamarosan siker is koronázta. 1969-ben hozták piacra a 256 bit tárkapacitású, ún. statikus RAM-ot, majd egy évvel később az 1024 bites, dinamikus RAM integrált áramkört. A dinamikus RAM megoldás különösen piacképesnek bizonyult, mivel az elemi információ lehető legegyszerűbb tárolása (tulajdonképpen elektromos töltések formájában) miatt az alkalmazott mikroelektronikai technológia adott színvonala mellett négyszeres kapacitású chipet tudtak kialakítani. Mindez tükröződött az 1 bitre eső, arányosan alacsonyabb árban, és még piacképesebbé tette a MOS memória technológiát.

A 70-es évekre a nagyszámítógépek és az egyre nagyobb teljesítményű mini- (megamini, szupermini) számítógépek széles körű vállalati elterjedése volt a jellemző. Mindez gyakorlatilag korlátlan felvevőpiacot jelentett a dinamikus RAM-ok gyártóinak, hiszen ezekhez a gépekhez egyre több és egyre nagyobb kapacitású memóriarendszerekre volt szükség. Éppen használt technológiai berendezéseiket így teljes kapacitással tudták üzemeltetni, viszonylag rövid idő, 3–4 év alatt át tudtak térni a következő technológiai generációra, és egyre jobb minőségű, egyre nagyobb termelékenységű berendezéseket tudtak üzembe helyezni. Az Intel 1982-ben bevezetett HMOS–III technológiája például már a hatodik volt a sorban, és a sebességteljesítmény szorzat két nagyságrenddel jobb volt, mint a legelső technológiáé.

Az egyedi áramkörök tervezési technológiája is korlátozás nélkül fejlődhetett, lehetővé téve, hogy minden három év után újabb, négyszeres tárkapacitású és gyorsabb elérési idejű RAM generáció kerüljön gyártásba. Így jelentek meg egymás után a 4, 16, 64 és 256 kbites dinamikus RAM-ok.

A számítógépgyártók „memóriaéhsége" még az egyes chipek kapacitásnövelésénél is erősebben növekedett, így az egyre többe kerülő gyártóberendezéseket, valamint az egyre növekvő tervezési költségeket a RAM chipek áraiban nem lehetett érzékelni.

A 80-as években ez azt jelentette, hogy az éppen legnagyobb tömegben használt dinamikus RAM generáció egyetlen chipjének ára, kapacitásától függetlenül, hozzávetőleg mindig 4 dollár körül alakult. Az egy bitre eső ár állandóan csökkent és a számítógépgyártók 2–4 évenként tudtak áttérni a következő generációra (lásd a *táblázatot*). Ez az áttérés jelentkezik az állandóan csökkenő memóriaárakban, amint az *ábrán* látható. A ciklikus piaci tendencia a továbbiakban is folytatódni fog, ehhez azonban a 80-as évek elejétől már a mikroprocesszor alapú számítógépek egyre bővülő termelése biztosítja a hátteret.

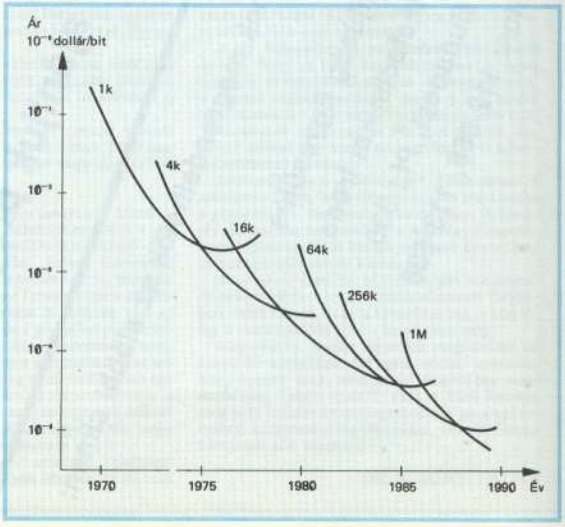
A mikroprocesszoros technika kezdetei

Egy japán zsebszámológépgyártó, a Bizcomp Corp. azzal kereste meg 1969-ben az Intelt, hogy az készítsen számára néhány nagy integráltságú céláramkört berendezéseihez. A tervezés már akkor sem került kevésbe, így az első mérlegelésnél az Intel el akarta vetni az üzleti ajánlatot. A későbbi felülvizsgálat során azonban az Intel egyik fiatal mérnökének, Marcian E. Hoffnak mentőötlete támadt. Azt javasolta, hogy készítsenek egy általános célú processzort, majd azt programozzák be úgy, hogy képes legyen támogatni az igényelt zsebszámológép-funkciókat. Az így kidolgozott és 1971 novemberében piacra hozott, Intel 4004 típusú mikroprocesszor még végletesen primitív volt. 4 bites adatokkal tudott műveletet végezni, mindössze 4,5 kbájt memóriát volt képes megcímezni, de már megjelentek benne az azóta is jellemző főbb architekturális elvek, mint például a belső sínszervezés.

Egy másik megkeresés eredményeképpen született meg 1972-ben az Intel első 8 bites mikroprocesszora, az Intel 8008. A Computer Terminal Corporation (jelenleg Datapoint) gyors, diszkrét elemekből (TTL) akart egy bitsoros processzort kialakítani, olcsónak tervezett képernyős termináljai programozott megvalósításához. A tároló verem, ún. stack egységet viszont az Inteltől kívánta beszerezni nagy integráltságú áramkör formájában. Az Intel ellenjavaslatára az egész processzort egyetlen chipen kezdték megvalósítani. A javasolt utasításkészletet további utasításokkal egészítették ki, 16 kbájt címzését alakították ki és a 8 bites adatkezelési képességet egy 8 bites mélységű processzor stack koncepcióval kombinálták. Az alkalmazott p-MOS technológia miatt az így elkészült processzor elég lassú volt (0,5–0,8 MHz ütemfrekvencia), hasonló jellegű alkalmazásra, tehát berendezések cél szerinti funkcióinak programozott (emulált) megvalósítására azonban már bátran lehetett ajánlani.

A 8008 átültetésekor az Intel második generációs, n-MOS technológiájára, a mikroprocesszor chipeket tervező Federico Faggin úgy találta, hogy érdemes lenne az egész processzor-architektúrát átterveznie. Így született meg az 1974 elején piacra került Intel 8080 típusú mikroprocesszor. A 2–3 MHz ütemfrekvenciájú processzor már 64 kbájt memóriát volt képes meg-

A dinamikus RAM áramkör tárkapacitása	1 kbit	4 kbit	16 kbit	64 kbit	256 kbit	1 Mbit
A mintapéldányok megjelenésének éve	1970	1973	1976	1979	1982	1985
A következő generációra való, ár szerinti áttérés éve	1975	1979	1983	1983	1988	–
Ár az áttérés pillanatában (10^{-5} dollár/bit)	200	30	7	3,5	1	–



címezni, és utasításkészletét sem néhány hét alatt, hanem sokkal alaposabban átgondolva alakították ki.

Faggin már ekkor úgy látta, hogy a 8080 architekturális képességeit (utasításkészlet, regiszterek stb.) továbbfejlesztve újabb processzort is ki lehet fejleszteni, ami már számítógép-építési célokra is tökéletesen megfelelne. Javaslatát azonban az Intel vezetői nem támogatták, mivel úgy vélték, hogy ennek nincs perspektívája, sőt ez már számítógépgyártó vevőkörük üzleti érdekeit is veszélyeztetné. Faggin eltávozott tehát az Inteltől, és az Exxon olajmonopólium pénzügyi támogatásával megalapította a Zilog céget. Így született meg 1976 márciusában a 8080-nal felülről kompatíbilis Z80 processzor.

Az Intellel konkurráló Motorola cégnél, a 6800-as mikroprocesszor fejlesztésében működött közre egy másik neves személyiség, Chuck Peddle. (Nem egy mikroszámítógépes vezéregyéniség keresztneve becézett alakjában fordul elő a mikrogépes sajtóban, így mi is ezt a gyakorlatot vagyunk kénytelenek követni.) A Motorola-nál Peddle néhány egész eredeti áramköri megoldás kialakításával tűnt ki. Így például az ő nevéhez fűződik a mikroprocesszoros gépek bemeneti/kimeneti alrendszerének gyors fejleszthetőségét támogató PIA (Peripheral Interface Adapter) koncepció és annak megvalósítása nagy integráltságú áramköri chip formájában. Számítógépes rendszerszemlélete egyáltalán nem meglepő, hiszen 1959-től kezdve, 11 éven keresztül dolgozott a General Electric számítógépgyártó részlegénél. Számítógépes múltja indíthatta őt arra, hogy munkatársaival együtt 1974-ben átmenjen a MOS Technology nevű, újonnan alakult céghez, és belefogjon egy igazán olcsó és architekturálisan igen hatékony mikroprocesszor, a 6502 fejlesztésébe.

Az 1975 második felében megjelent 6502-t valóban igen olcsón kínálták. Kiskereskedőnél mindössze 25 dollárba került, szemben a többi mikroprocesszor kb. 200 dolláros árával. A dinamikus RAM áramkörök generációit bemutató táblázatunkra pislantva láthatjuk, hogy éppen ebben az időben lesznek előnyösek a 4 kbites DRAM áramkörök, vagyis néhányszor tíz dollárért már 4 kbájt, illetve 8 kbájt kapacitású memóriához is be lehetett szerezni az áramköröket. Peddle még más chipeket (például a PIA egyik továbbfejlesztett változatát) is kidolgozott a 6502-höz, így minden készen állt ahhoz, hogy akár a számítógép-építő hobbisták készíteni tudjanak maguknak személyes használatú mikroszámítógépet. A hobbizmus pedig, mint látni fogjuk, elvezetett a személyi számítógéphez.

Hobbiból lett házipar

Az első, nagyobb teljesítményű mikroprocesszorokat már általános célú gépbe is érdemes volt beépíteni, de nem volt könnyű megtalálni sem a megfelelő piacot, sem pedig az ennek megfelelő számítógép-konstrukciót. A francia R2E által kifejlesztett első mikroszámítógép nem ért el azonnali, átütő piaci sikert. Az USA-ban is több kísérlet volt 1974 során (például az Intel 8008 alapú Mark–8 készlet), amelyeknek szinte alig volt hatásuk az események további alakulására.

Az igazi mikrogépes piacot a Micro Instrumentation and Telemetry Systems (MITS) nyitotta meg 1975 januárjában, amikor Altair 8800 típusú gépének hirdetése megjelent a Popular Electronics című magazin címlapján. A MITS-et 1970-ben alapították New Mexico állam Albuquerque nevű városában, egy Ed Roberts nevű úriember garázsában. A cég elsősorban olyan kísméretű elektronikus berendezéseket gyártott sikerrel, amelyek játékrakétákba és repülőmodellekbe építve a helyzetjelzésre, a kényszerleszállási ejtőernyő működtetésére és fényjelzők vezérlésére szolgáltak. Nem volt tehát a cégtől idegen az elektronika hobbi célokra való alkalmazása, és ez a továbbiak szempontjából döntőnek bizonyult.

A Roberts által tervezett Altair 8800 egyáltalán nem hasonlított külső megjelenésében a ma ismert mikroszámítógépekre. Előlapján az akkori miniszámítógépekéhez hasonló kapcsoló- és lámpasor helyezkedett el. Belsejében viszont egy nyitott, jogilag nem levédett, egyszerű sínrendszert alakított ki Roberts, így eleve biztosítva volt a moduláris bővíthetőség nemcsak az eredeti gyártó által szállított kártyákkal, hanem más gyártók adott feladatra kialakított kártyáival és az ezekhez csatlakozó részegységekkel is. Robertset persze inkább az vezette erre a megoldásra, hogy a hobbisták számára ezzel is vonzóvá és minimális kiépítésű alapkészlet (kit) formájában egyúttal elérhetővé is tegye termékét. A vevő által összeszerelendő alapkészletet 395 dollárért hirdette meg a MITS, míg az összeszerelt változatot 621 dollárért árulták.

Összehasonlításként érdemes megemlíteni, hogy az Altair 8800 Advanced Accouting/Engineering Systemnek nevezett teljes kiépítésű rendszert 10 500 dollárért hoztak forgalomba. A valóban olcsó Altair 8800 alapkészletből több ezret adott el a MITS 1975 év során, és így valóban megalapozta azt a lavinaszerű folyamatot, ami azóta is jellemzi a mikroszámítógépek széles körű elterjedését.

1975 másik meghatározó eseménye volt az első számítógép-építő klubok megalakulása. Ekkoriban még igencsak szűkében voltak a hobbisták a mikroelektronikai elemek alkalmazásával és az első gépek „továbbépítésével" kapcsolatos információknak, rászorultak egymás szakismeretére és segítségére.

A kaliforniai Menlo Parkban 1975 márciusában megalakult Homebrew Computer Club (HCC) volt talán a legelső a klubok között. Ennek a csoportosulásnak mégsem ez adja meg a hírnevét, hanem az a sajátos szerep, amit a mikroszámítógépes ipar kibontakoztatásában betöltött. A Gordon French nevű úr garázsában megrendezett első összejövetelen és az első év további összejövetelein szinte kivétel nélkül olyanok vettek részt, akik vállalkozóként, konstruktőrként vagy inkább mindkét minőségben jelentős szerephez jutottak a későbbi mikroszámítógépes ipar kialakulásában.

Roger Melen és Hary Garland, a HCC tagjai és a Cromemco cég alapítói hozták forgalomba az első, Altair sínrendszerbe dugaszolható kártyát. Dazzlernek (káprázтatót, de más értelemben meghökkentőt, elrémítőt és elképesztőt is jelent) nevezett termékük színes grafika megjelenítését tette lehetővé a tévéképernyőn, ezért az első színes grafikai kártyának is tekinthető.

Mivel akkoriban még meglehetősen kevés számítógépes szaküzlet volt, a Cromemco elsősorban postai csomagküldő módszerrel forgalmazott. Az első szaküzletek viszont sikerrel használták az érdeklődés felkeltésére. Az egyik nevezetes sztori szerint New York város éjszakai autósait valósággal hipnotizálta a Dazzlerhez szállított Kaleidoscope nevű program segítségével generált véletlenszerű mintázat, amit az üzlet kirakatában elhelyezett tévé képernyőjén láthattak. Az ezért keletkezett forgalmi dugót csak úgy tudta megszüntetni a rendőrség, hogy éjnek idején megkereste a tulajdonost, és kikapcsoltatta a veszedelmes készüléket.

Melent és Garlandot egyre többen követték, és egymás után jöttek létre a garázsokban és más, a házhoz tartozó helyiségekben működő kisvállalatok. Egy sajátos háziipar (cottage industry) alakult ki tulajdonképpen, ami egyben meghatározta a mikroszámítógépes ipar máig fennálló, széles körű kooperáción alapuló működési jellegét.

Az ilyen kezdetet persze nemcsak a kiegészítő kártya- és részegységgyártók határozták meg, hanem azok az újonnan alapított vállalatok, amelyek mikroszámítógépük konstrukcióját az Altair 8800 nyitott sínrendszerére alapozták. Ilyenek voltak az IMSAI és a Processor Technology. Az utóbbinak az alapítója és főkonstruktőre is HCC-tag volt.

Ezek az újabb mikroszámítógépek még csak kismértékben javítottak az eredeti Altair konstrukción. A minigépszerű elölapot felváltották az állandó (ROM) memóriába beírt programból (firmware) való működtetésre, amihez írógépszerű billentyűzet állt rendelkezésre. Az Altair sínrendszert azért hagyták meg, mert az első időszak számítógép-építő vevői már ismerték azt, a későbbiekben egyre nagyobb számú többi vevőnek pedig gépe komplettálása szempontjából volt az előnyös, hogy a kiegészítő kártyákat az alapberendezés-gyártótól függetlenül is be tudta szerezni. Valamennyi gyártónak is előnyös volt ez a megoldás, mivel csak egy szűk területre kellett szűkös erőforrásaikkal koncentrálniuk, és mégis számíthattak arra, hogy széles körű piaca lesz termékeiknek.

Ugyanebben a kezdeti időszakban alakultak ki azok a kereskedelmi és árubemutató helyek, amelyek azóta is a leginkább meghatározóak. Az egyre terjedő kiskereskedelmi számítógépes szaküzletek a lehető legrugalmasabb módon tudták közvetíteni a bővülő vevőkör felé az egyre szélesebb áruválasztékot. Megrendeléseikkel egyúttal hatékonyan befolyásolták az újabb és újabb gyártói törekvéseket. A hobbistáknak találkozóhelyet biztosítottak, és a klubokhoz hasonlóan, az információ és tapasztalatcsere sajátos fórumává lettek. A boltok propagandája révén ezrek és ezrek lettek az új számítógépes őrület megszállottjai. A klubok, a háziipar és a gyarapodó bolthálózat így további, még gyorsabb mozgásra ösztönözte a megindult mikroszámítógépes lavinát, miközben egymást is erősítették.

Hamarosan szükségessé váltak az országos méretű kiállítások és termékbemutatók, ahol a gyártók a kereskedőknek, valamennyi érdeklődőnek és egymásnak be tudták mutatni új termékeiket. Először a keleti parton rendeztek ilyen eseményt 1976-ban (East Coast Computer Show), majd a rivális nyugati parton egy évre rá (West Coast Computer Faire). Az utóbbi megszervezésében, talán mondanunk sem kell, a HCC egyik első tagja, Jim Warren járt az élen.

Az első keleti parti bemutató arról is nevezetes, hogy itt kapott új nevet a már több gyártó által is használt Altair sínrendszer. Hogy ne kelljen ezeknek az új gyártóknak ingyen propagandát biztosítani a velük versenyben álló Altairnek, azt találták ki, hogy S–100 sínrendszernek nevezzék, mivel 100 érintkezős csatlakozási felületeket írt elő kártya szinten. Ekkor még nem beszélhettünk igazi szabványról, mivel hiányzott pontos és mindenre kiterjedő meghatározása, de kb. két évvel később, a második nyugati parti számítógépvásáron már erre is sor került.

A 80-as évek elején országos ágazati szabványként (IEEE–696) is megjelent az első, berendezésgyártótól független ipari szabvány, bizonyítva azt a jelentős

szerepet, amit a mikroszámítógépes iparban betöltött. A lavina további erősödését más berendezésgyártótól független, de facto szabványos megoldások is elősegítették, elsősorban az alapszoftver területén.

Az olcsó, szabványos alapszoftverek

Két, a középiskolát éppen csak hogy elvégzett fiatalember, Bill Gates és Paul Allen az Altair 8800 megjelenésekor úgy érezték, hogy most jött el az ő idejük. Már korábban is foglalkoztak a gondolattal, hogy a 8008 mikroprocesszorra egy BASIC rendszert készítsenek, de akkor úgy találták, hogy annak utasításkészlete még nem elégséges a feladathoz. Annak ellenére, hogy nem volt Altair gépük, nem riadtak vissza magától a feladattól, mivel már több éves szoftverfejlesztő múlt állt mögöttük üzleti értelemben is. Az Intel 8080 kézikönyv alapján egy 8080-at utánzó (szimulátor) programot írtak a kis cégük számára elérhető PDP–10 nagyszámítógépre, majd Gates megtervezte a 8080 BASIC értelmező (interpreter) programot, amit a szimulátoron próbálhattak ki először. Ezután hívták fel a MITS-et, ahol mindjárt rájöttek, hogy a két fiú nem a levegőbe beszél, és repülőgépen székhelyükre vitették őket.

Ilyen előzmények után jelent meg a piacon az első mikroszámítógépes BASIC rendszer. Ez nagyon kompakt változat volt, mivel mindössze 4 kbájt memóriára volt szüksége, amiből 3,2 kbájtot foglalt el maga az interpreter. Igen komoly szoftverfejlesztési teljesítmény, ha meggondoljuk, hogy ez a tömör rendszer még lebegőpontos aritmetikával is rendelkezett.

Hobbisták között tehát villámgyorsan terjedt Gates és Allen 4 k-s BASIC-je, piacképes volt adaptációja más 8080 alapú gépekre is, illetve egyre fokozódó igény jelentkezett az egymással felülről kompatibilis, de önmagában egyre többet tudó BASIC rendszerek sorozatára. Ebből nőtt ki a Microsoft nevű független szoftvergyártó vállalat, élén a még ma is roppant fiatal egyik alapítóval.

Az Altairt követő első gyártók egyike volt az IMSAI. Floppy háttértárral is felszerelhető gépének első példányait még azzal szállította le első vevőinek, hogy az operációs rendszert majd valamikor később biztosítja. Az IMSAI egy aprócska kaliforniai céget, a Digital Research-t kérte fel, hogy gépéhez kidolgozzon egy megfelelő operációs rendszert. A cég alapító tulajdonosa, Gary Kildall, főállásban a Naval Postgraduate School tanára volt, és már a 4004 megjelenése óta készített alapszoftvereket az Intel mikroprocesszor alapú rendszereire. Az Intel megbízásából ő fejlesztette ki például a PL/M nevű nyelvi rendszert, és ennek során, mintegy melléktermékként, már kifejlesztett egy lemezes operációs rendszert. Ezt a rendszert tudta igen gyorsan adaptálni Kildall az IMSAI gépekre. Később rájött, hogy rendszerének gépfüggő részeit egy külön részbe, a BIOS-be (Basic Input–Output System) zárva, más 8080 alapú gépekre is könnyen átvihető lesz az operációs rendszer.

Az első nyugati parti számítógépes vásáron a vásár alapítója, Jim Warren arra beszélte rá Kildallt, hogy szoftvercsomagot készítsen rendszeréből. Így a hobbisták maguk tudták adaptálni azt mikrogépük konkrét hardverrendszerére. Ez akkori szemmel nézve óriási piacot jelentett, mivel jellemző volt, hogy a hobbisták maguk komplettálták mikroszámítógépeiket például háttértárrendszerekkel, így a megfelelő operációs rendszerről is nekik kellett gondoskodniuk. A CP/M 1.3-nak nevezett első változat mindössze néhány száz dollárba került, így egyre nőtt az érdeklődés a vevőkör, mígnem a CP/M de facto ipari szabvány nem lett. Elterjedésében fontos tényező volt a magasabb szintű szoftverelemek (például alkalmazási szoftver) irányában mutatott szabványos csatlakozási felülete is.

Lavinaszerű folyamatként jelentkezett az, hogy érdemes volt egyéb rendszer- és alkalmazási szoftvert erre a hordozható operációs rendszerre írni, minek következtében még jobban növekedtek az operációs rendszer eladásai, ezáltal viszont még fokozottabban érdemessé vált a szoftvereket ehhez az operációs rendszerhez igazodva fejleszteni. Erre a folyamatra rátalálva válhatott a kibontakozó mikroszámítógépes szoftveripar másik „óriásává" a Digital Research cég.

Az Intel innovatív mikroelektronikai elképzelései hamar elvezettek egy sajátos konstrukciójú számítógépes piachoz, jóllehet a cég eredeti elképzelései között ilyen egyáltalán nem szerepelt. A sajátos konstrukciót az jellemezte, hogy az egyén vált a mikroszámítógép vásárlójává, a hardver és a szoftver iparilag szabványos csatlakozási felületei biztosították a lehető legalacsonyabb komplettálási árat, valamint a lehető legkedvezőbb bővítési lehetőséget, és mindez az érdeklődők egyre nagyobb tömegének tette elérhetővé a számítógépet.

Ez tehát a mikroszámítógépes lavina első hullámának lényege, aminek értelemszerűen el kellett vezetnie a még nagyobb hatású személyi számítógépes konstrukció megjelenéséhez.

NACSA SÁNDOR
(Folytatjuk)

Sakkprogramozás

Specializált számítógépet vagy programot?

Milyen számítógépet vásároljunk? sorozatunk keretében közbevetőleg egy olyan kérdéssel foglalkozunk, amely – amint a mind gyakrabban hozzánk intézett kérdésekből is kitűnik – sokakat foglalkoztat, akik számítógép ellen szeretnének sakkozni. A kérdés így hangzik: „Mit vásároljak inkább, sakk-számítógépet, vagy vegyek például egy Sinclairt vagy Commodore 64-et és ahhoz sakkprogramot?". E kérdés természetesen belevág témakörünkbe, hiszen amikor az olvasónak vásárlási döntésében kívánunk segítséget nyújtani, éppúgy gondolunk a sakkszámítógépekre, mint a személyi számítógépekre írt sakkprogramokra – ha ez sorozatunk címében eddig kifejezetten nem is szerepelt.

Két piac

Talán meglepi az olvasót, mégis jellemző az elektronikus sakkpartnerek mai piaci helyzetére, hogy az iparilag fejlett országok nagy kínálatú boltjaiban, áruházaiban másutt kell keresni a sakk-számítógépeket és másutt a programokat.

Érdekes kísérletről olvastunk a Computerschach und -Spiele című NSZK-beli folyóirat egyik legutóbbi számában. Egy szakember végigjárt egy sor szaküzletet, amelyben személyi számítógépeket és ezekre írt programokat, főként játékokat árusítanak, és sakkprogramokat keresett. A legtöbb helyen egyet sem vagy csak nagyon kevés típust talált, több esetben használati utasítás nélkül; szakszerű felvilágosítást pedig jóformán sehol sem kapott. (Itt hívjuk fel a figyelmet: senki, soha ne vásároljon semmilyen számítógépet vagy programot, akár újonnan, akár használtan a kezelési útmutató nélkül!) A magyarázat egyszerű: a személyiszámítógép-kereskedelemnek a sakkprogram valahol a perifériáján szerepel. A sakkszámítógép viszont önálló, igen elterjedt üzletág; csaknem valamennyi nagy áruházban és számos szaküzletben kaphatók, nagy választékban, és szakszerű felvilágosításban sincs sehol hiány.

Itt mutatjuk be olvasóinknak az említett folyóiratból vett jegyzéket, amely a forgalomban levő, különböző személyi számítógépekre írt programok legtöbbjét tartalmazza, sok esetben DM-ben megjelölt árukkal együtt. Ez az összeállítás hasznos információforrás, de számolni kell azzal, hogy aki egy bizonyos programot szeretne megvásárolni, nem könnyen találja meg.

Játékerő-összehasonlítás

Aki saját személyi számítógépéhez kíván sakkprogramot vásárolni, viszonylag szűk választékot talál, hiszen ugyanahhoz a géphez nem áll tucatnyi program rendelkezésre. A választáskor ugyanazokat a szempontokat kell figyelembe venni, mint a sakk-számítógépek esetében: a játékerőt, a lépések megjelenítésének módját, a program különleges funkcióit, s bizonyos mértékben a kivitelt, az esztétikai kívánalmakat is, ami itt például a tetszetős grafikai megoldásban vagy a használati utasítás áttekinthető kivitelezésében, a kazetta vagy diszkett és a könyvecske csomagolásának vonzerejében rejlik. A tényezők azonban más súllyal esnek latba, mint a specializált személyi számítógépek esetében.

A kielégítő játékerő természetesen egyformán fontos, bármilyen elektronikus partnerrel szemben küzdünk. Meg kell mondani, hogy a legjobb programok – például a *Sargon III* különféle verziói, amelyeknek alkotói a rovatunkban már többször említett Kathe és Dan Spracklen – sem érik el a mai kiemelkedő sakk-számítógépeket.

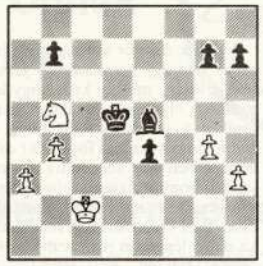
A múlt év szeptemberében Glasgow-ban tartott, 4. mikro-VB résztvevői között a Sargon III programnak három, az *Apple IIe*, az *Apple/Macintosh* és az *IBM* mikroszámítógépekre írt verziója szerepelt, s kielégítő eredményt értek el, de elmaradtak a legjobb specializált számítógépektől. Érdemes megjegyezni, hogy az Apple verziókhoz gyorsító programot alkalmaztak; amely az eredetileg 1 MHz sebességű processzor teljesítményét három-négyszeresére fokozza.

PROGRAM	SZERZŐ	ÉV	TÁR	HORDOZÓ	SZÁMÍTÓGÉP	ÁR (DM)
Microchess 1.0	Jennings	78	1 k		AIM 65	
Microchess 1.5	Jennings	78	4 k		TRS 80	
Microchess 2.0	Jennings	78	8 k	kazetta	PET 2001	75,—
Microchess 2.0	Jennings	78	16 k	lemez	Apple II	
Sargon I	Spracklen	78	8 k	lista	Jupiter 3	50,—
Sargon I	Spracklen	78		lemez	TRS 80	
Sargon I	Spracklen	78		lemez	Apple II	
Sargon II	Spracklen	79		lemez	TRS 80	
Sargon II	Spracklen	79	24 k	lemez	Apple II	
Sargon II	Spracklen	83		ROM	VC 20	59,—
Sargon 2,5	Spracklen?			kazetta	Sharp	62,—
Sargon III	Spracklen	83	48 k	lemez	Apple II	199,—
Sargon III	Spracklen	84	300 k	lemez	Macintosh	
Sargon III	Spracklen	84		lemez	Liza	
Sargon III	Spracklen	84		lemez	IBM PC	
Chess 7.0	Atkin	84	48 k	lemez	Apple II	249,—
Chess 7.0	Atkin	84	48 k	lemez	Atari	249,—
How about...	Atkin	84	64 k	lemez	Apple II	105,—
Mychess A	Kittinger			lemez	Cromenco	
Mychess A	Kittinger			lemez	Osborne	
Mychess B	Kittinger			lemez	Cromenco	
Mychess B	Kittinger			lemez	Osborne	
Schachmeister	Levy	81		ROM	TI 99/4A	99,—
Boss	Schäfer	82		lemez	VC 20	
Grandmaster	Schäfer	83		kazetta	C64	79,—
Grandmaster	Schäfer	83		lemez	C64	79,—
The Mate		82	48 k	lemez	Apple II	798,—
Chess CXL 4009		79		ROM	Atari	119,—
Chess		83		ROM	Atari	139,—
Sfinks	Fink				TRS 80	
Cyrus	Int.Softw.	81		ROM	Dragon 32	129,—
Caissa	Wrensch	84		lemez	C64	79,—
Colossus 2.0	Bryant	84	26 k	kazetta	C64	65,—
Chess			48 k	kazetta	Oric	49,50
Chess		83		kazetta	Acorn B	
White Knight	Bryant			kazetta	Acorn B	
Alybadix (Prb)	Blom			lemez	Apple II	
Mate 4F (Prb)	Korhonen			lemez	Apple II	
Videopac C7010	Rens	82		ROM	G 7400	399,—
Schach		81	16 k	kazetta	ZX81	39,50
ZX Schach 2			16 k	kazetta	ZX81	65,—
Spectrum Chess		82	48 k	kazetta	ZX-Spectrum	39,50
Chess The Turk	Hutchby	82	48 k	kazetta	ZX-Spectrum	49,—
The Chessplayer	Wren Hilton	82	48 k	kazetta	ZX-Spectrum	39,50
Cyrus IS	Int.Softw.	81			ZX-Spectrum	
Petchess					CBM 8032	
Chess Bluebush					IBM PC	
Spoc. The Chessm.				lemez	IBM PC	

Jó alkalmat nyújtott az összehasonlításra az angol *Martin Bryant* alkotta *Colossus* program legújabb, ugyancsak Apple-re írt, 2,1 indexszel ellátott verziójának szereplése is. Ennek teljesítménye elmaradt a Sargonokétől. A Colossus 2,1 – amikor e sorokat írjuk – még nincs a piacon, de nálunk is minden bizonnyal sokan ismerik a *Commodore 64*-re írt, *Colossus 2* programot, amely az 1984. év számítógépére készült, ez ideig talán legjobban sakkozó program. Ennél pedig – amint alkotója maga is megerősítette – az Apple verzió, már csak a számítógépek teljesítőképessége közötti különbség miatt is, feltétlenül jobb.

Sietve hozzátesszük: a Colossus programok egy közepes játékerejű sakkozó igényeit kielégítik; a Commodore-tulajdonosoknak e program megvételét nyugodtan javasoljuk. Igen érdekes volt Glasgow-ban a személyi számítógépekre írt programok játszmáit figyelemmel kísérni. A sorsolás szeszélye két Apple-t is szembeállított egymással, a Sargon III/Macintosh, illetve a Colossus 2,1 programmal. A játszma színvonalas, mindkét fél részéről szinte hibátlan volt. A végjátékban Sargon helyzete némileg előnyösebbnek tűnt, de ennek érvényesítéséhez a gépek végjátékismeretéhez képest alapos tudásra volt szükség.

Colossus 2,1/Apple II – Sargon III/Apple/Macintosh



Hadállás sötét 49. lépése után

Következett *50. Hc3+, Kd4!* (A program „tudja", hogy a királyával a gyalog elé kell kerülnie.) *51. h4* (A király behatolása ellen nincs védelem, mert 51. Hd1-re e3! után e2 megállíthatatlan.) *51. – Ke3 52. g5?* (Tipikus számítógéphiba, amelyet azonban egy igazán jó program ma már nem követ el. Nem „látja", hogy a következő csere után az *e* gyalogot nem lehet feltartóztatni. Ezért okvetlenül Hd5+ kellett, s csak Kf3 után g5. Sötét akkor is nyer, de még nincs könnyű dolga.) *52. – Fxc3* (A program fejlettségének tanújele, hogy ez a csere, amely két lépéssel előbb még bizonytalan kimenetelű volt, ebben a helyzetben azonnal nyerésre vezet.) *53. Kxc3, Kf2 54. Kc4, e3* és *sötét nyer.*

A személyi számítógépek előnyei

Természetesnek kell tartanunk, hogy ha egy hardver kifejezetten a sakk céljaira készül, magasabb játékerőt lehet vele elérni; az általános számítógépek tervezői nem gondolhatnak olyan különleges szempontokra, amelyek előmozdítják a sakktudást. Ugyanakkor azonban a személyi számítógépek rendelkeznek más adottságokkal, amelyeket a sakkra specializáltak nélkülöznek.

Első helyen a képernyőt kell említeni. Aki hozzászokik a sakkábrákhoz, igen élvezi az esztétikusan megtervezett diagramok útján történő játékot. Ehhez persze ismerni kell az algebrai sakkjelölést is, mert a lépéseket betáplálni csak a billentyűzeten lehet, kivéve, ha a készülékhez botkormány is rendelkezésre áll, amellyel igazán élvezetes a bábukat a kurzor útján ide-oda tologatni. A színes képernyő csak fokozza a játék nyújtotta esztétikai hatást.

Ennél fontosabb, hogy a képernyő nagy felülete sokkal jobb lehetőségeket kínál különféle információk közlésére, mint egy sakkgép, és annyiszor lehet egy-egy állásban különböző célokra felhasználni, ahányszor a programozó kívánja. Csaknem minden fejlett program részletes betekintést enged például abba, hogy mely lépést, lépéseket elemzi. A sakkszámítógép kívánságra szintén megmutatja az éppen vizsgált lépést, de egyszerre csak egyet; többre nincs fizikai lehetőség, mivel ezt rendszerint a diódák kigyulladásával teszi. A személyi számítógép sorra ki tudja írni, hogy mely lépéseket, változatokat – akárhányat! – elemzi. Persze bármikor megadja a hadállás értékszámát is (amelyről a használati utasítás bővebben tájékoztat), és hogy hányadik fellépésnél tart; mindezeket egyszerre, egy képernyőkijelzéssel. Ezt az adottságot a jó sakkozók is nagyon szívesen fogadják.

Összefoglalás

A bevezetőben felvetett kérdésre tehát azt válaszolhatjuk, hogy aki nem arra helyezi a legfőbb súlyt, hogy elektronikus partnere a lehető legfejlettebb sakktudással bírjon, és főként ha készülékével nemcsak sakkozni akar, vásároljon nyugodtan sakkprogramot személyi számítógépéhez, jól fog szórakozni vele. Hívja segítségül az itt közölt táblázatot, de levélbeni megkeresésre szívesen adunk további felvilágosítást, tanácsot bárkinek, aki sakk-számítógépet vagy sakkprogramot kíván vásárolni.

DR. LINDNER LÁSZLÓ

COMMODORE

- VC 20 RAM bővítő 8–16 k,
- C 64 és VC 20-ra írt szótárprogram,
- C 64 EPROM égető,
- C 64 szoftver.

ELPROTERV GMK

Telefon: 805-869

LÉPÉSRŐL LÉPÉSRE

Tud-e a sakkprogram tanulni?

Az ember folyamatosan tanul a hibáiból: minden elvesztett játszmából leszűr bizonyos következtetéseket. Keresi a vesztés okát, és további játszmáiban igyekszik az elkövetett hibát elkerülni. Jó lenne valahogy ezt a képességet a számítógépbe is beépíteni, hiszen, mint látjuk, pusztán a számítógép lehetőségeinek, gyorsaságának kihasználásával nem tudunk verhetetlen sakkprogramot készíteni.

Az ember, amikor hibáiból tanul, gazdag fogalomkörben mozog, a sakkról meglevő igen sok ismeretébe építi be azokat a tanulságokat, amik a jövőben a hasonló vereségek elkerülésére vezethetnek. A jelenlegi sakkprogramok azonban, mint az eddigiekből kiderült, rendkívül keveset tudnak magáról a sakkról.

Az ember egy vereségből levonhat például egy olyan következtetést, hogy „ebben a változatban mielőtt a királyszárnyon támadást indítanék, le kell zárni a c vonalat", de milyen következtetést vonhat le egy sakkprogram?

Az előző mondat egyetlen fogalmát sem ismeri. Ha megpróbáljuk az előző mondatot megfogalmazni azokkal az eszközökkel, amelyekkel a sakkprogram dolgozik, legfeljebb valami olyasmit állíthatunk meg, hogy „az adott partiban a saját figuráimhoz rendelt mozgékonyságérték túl magas volt, a király biztonságát kifejező szám pedig túl alacsony, ez volt a hiba oka".

Ha ezt a hibát felismertük, megtehetjük, hogy a megfelelő pontértékeket a kiértékelő függvényben módosítjuk, de itt nem a program tanult, hanem mi, aminek eredményeképpen a programot átírtuk. Ha a program tanulna, akkor magától, automatikusan képes lenne a fenti következtetést levonni, és a módosítást végrehajtani.

Az első probléma az, hogy hogyan ismerje fel a program, mikor követte el a hibát. Ez többnyire még az embernek is problémát okoz, gazdag fogalomkincsével együtt; a programnak pedig szűkös eszköztára segítségével kell megfogalmaznia olyan kritériumokat, amelyek alapján egy lépést utólag hibásnak minősíthet.

Vegyük először a legegyszerűbb esetet. Ha egy program egy játszmát elveszít, mindig kell, hogy legyen egy olyan pont a játszmában, amikor a program már maga is tudja, hogy veszíteni fog, de az előző lépésben még nem tudta. Ez azt jelenti, hogy a vizsgált fában a minimax algoritmus a fa kiinduló csúcsához vesztő értéket rendel, de az előző lépésben még nem vesztő értéket rendelt. Azt, hogy ilyen pont biztosan van a játszmában, az bizonyítja, hogy a kiinduló állásban, a játszma kezdetén, a program még nyilván nem értékeli vesztettnek az állást, egy lépéssel a matt előtt pedig már biztosan annak értékeli.

Ezen a ponton első közelítésre megkapaszkodhatunk: minősítsük hibának azt a lépést, amit akkor tett, amikor még nem tudta, hogy veszíteni fog, de amire az ellenfél olyan lépéssel tudott válaszolni, hogy azután a program saját értékelése radikálisan megváltozott, és már ő is tudta, hogy veszíteni fog. Természetesen semmi sem garantálja, hogy tényleg ez a lépés volt a hiba, hiszen lehet, hogy a program értékelései nem voltak pontosak, és tudtán kívül már jó néhány lépés óta vesztő állásban volt.

Azt könnyen megakadályozhatjuk, hogy a program a hibásnak minősített lépést máskor is megismételje. Ehhez elegendő, ha feljegyezzük a program életében lejátszott összes elvesztett játszmákat, és mindegyikben megjelöljük a hibásnak minősített lépést. A továbbiakban ezt a lépést örökre letiltjuk: úgy kezeljük, mintha az adott állásban szabálytalan lépés lenne.

Ennek a megoldásnak van egy távolabbra mutató haszna is: előfordulhat, hogy a program akkor is el fogja kerülni a továbbiakban a hibát, ha azt nem konkrétan a hibásnak minősített lépés okozta, hanem valamelyik korábbi. Eléggé valószínű ugyanis, hogy a minimax algoritmus éppen a hibásnak minősített lépés következményeit értékelte hibásan, és már jóval korábban emiatt a változat miatt döntött egy lépés mellett. Ha ez így történt, akkor a program ezentúl a jóval korábban elkövetett hibát is el fogja kerülni, pusztán a letiltott későbbi lépés miatt.

A megoldást mégsem érezzük kielégítőnek: idővel nagyon elszaporodnak a vesztett játszmák, és valamennyit végignézni minden egyes lépéslehetőség megvizsgálásával rendkívül időigényes lenne. És különben sem elegáns a megoldás.

Egy ötlettel azonban lényegesen javíthatjuk a fenti megoldás hatékonyságát, sőt általánosabbá is tehetjük. Az ötlet a következő: ne magukat a lépéseket figyeljük, hanem azokat a számokat, amiket a program a minimax algoritmus végrehajtásánál végeredményként kapott, tehát amiket a program a lépések kiválasztásánál az éppen adott álláshoz rendelt. Ezek a számok nagyon sokfélék lehetnek: egy játszma szinte biztosan egyértelműen jellemezhető pusztán azzal, ha sorban felsoroljuk ezeket a számokat. (Ez nem azt jelenti, hogy csupán a számok alapján könnyű a játszmát rekonstruálni, hanem csak azt, hogy lehetséges.) Sőt, ha csak az utolsó néhány kiértékelőfüggvény-értéket adjuk meg, szinte biztos, hogy már az is egyértelműen jellemzi a játszmabeli szituációt.

Minél finomabb a kiértékelő függvény, annál biztosabb, hogy ez a helyzet. Ha például a kiértékelő függvény annyira durva, hogy csak az anyagi egyensúlyt számolja, akkor a rekonstrukció nem is végezhető el. A BELLE programnak az előző részben bemutatott kiértékelő függvénye egyszerűsége ellenére is szinte mindig elég a rekonstrukcióhoz.

Megtehetjük, hogy nem a konkrét, hibásnak minősített lépést tiltjuk le mindörökre, hanem azt, hogy az a néhány szám, amely a hibásnak minősített lépés megtételénél az utolsó néhány kiértékelőfüggvény-értéket jelezte, valaha is újra előálljon. Az eredmény lényegében ugyanaz, mintha a konkrét lépést tiltottuk volna le. Amit kockáztatunk, az annyi, hogy esetleg egy egész más helyzetben is előáll ugyanez a néhány szám, és a letiltás arra az esetre is érvényes lesz, pedig ott esetleg éppen az így letiltottnak tekintett lépés lenne a helyes. Ez a kockázat azonban elenyészően kicsi.

A letiltás viszont ténylegesen általánosabb, mintha csak a konkrét játszma konkrét hibás lépését tiltottuk volna le: például a program ezentúl akkor sem fogja elkövetni a hibát, ha a kritikus állás előzetes lépéscserékkel áll elő. Sőt szerencsés esetben még akkor sem, ha az állás nem pontosan azonos az elvesztett játszmabelivel, csak nagyon hasonló hozzá.

A bemutatott „tanulási" algoritmus még így is meglehetősen durva. Inkább arra hasonlít, amikor az ember elhatározza, hogy „ezentúl nem játszom ezt a változatot", mint a korábban leírt finomabb fajta tanulságokra.

Érdemes egy momentumot kiemelni, aminek a továbbgondolásával sokkal finomabb tanulási eljárásokhoz juthatunk. Hogy mi lehetett a hibás lépés, abból vettük észre, hogy a kiértékelő függvénynek a minimax algoritmus utáni végső értéke egyik lépésről a másikra ugrásszerűen lecsökkent. Ez így már jóval általánosabb, mint amennyit kihasználtunk belőle: egy ilyen radikális csökkenés mindig azt jelzi, hogy a kiértékelő függvény pontatlan volt. Az ellenfél tudott olyat lépni, ami után a kiértékelő eljárás (persze most már legalább egy lépéssel tovább tekintve előre) lényegesen más eredményt ad, mint előzőleg.

Ilyenkor a program megteheti, hogy utólag, „post festa" megvizsgálja, hogyan kellett volna a kiértékelő függvény egyes komponenseinek a súlyát úgy módosítani, hogy a két érték egymáshoz közelebb kerüljön. Jó esély van arra, hogy egy ilyen módosítás után a kiértékelő függvény pontosabb lesz, vagy legalábbis stabilabb, bár az eredményt elvileg semmi sem garantálja.

Ilyen elven működik például az amerikai A. L. Samuel dámát játszó programja, ami tíz év alatt úgy feljavította saját kiértékelő függvényét, hogy közepes erejű dámajátékosból mestererejűvé vált: legyőzte Michigan állam dámabajnokát, bár az Egyesült Államok bajnokától kikapott.

A most leírt tanulási elveknek, mint szinte minden gondolatnak a sakkprogramozás területén, számtalan változata, finomítása létezik. Valamennyi ténylegesen tanuló eljárás, abban az értelemben, hogy a programok saját játékukat változtatják, és szerencsés esetben ettől tényleg erősebbekké is válnak. Ahhoz azonban, amit az ember ért tanuláson, nem sok közük van.

Az ember a tanulás során nemcsak az algoritmusának egyes paramétereit képes megváltoztatni, hanem magát az algoritmust is: gondolkodása, egész sakkszemlélete változik, ahogy tanul és ahogy egyre erősebb sakkozóvá válik. A következő folytatásokban azt fogjuk vizsgálni, hogy mi az, amit az ember tud, de a sakkprogramok nem tudnak, és mi teszi minderre képessé az embert.

MÉRŐ LÁSZLÓ

a KERSZI

bemutatja Önnek
a SZATIR-t

a Személyi Állomány Tervezési és Információs Rendszer

elsődleges feladata az, hogy mérsékelje a személyzeti munkát végzők nagy adminisztratív megterhelését, lehetővé tegye a nyilvántartások kis helyen történénő tárolását, gyors visszakeresését, illetve módositását.

a SZATIR – jelenlegi állapotában –

a programot is tartalmazó lemezen legfeljebb **1000** személy adatait kezelheti. Egy dolgozóról **19** adatot tart nyilván. A dolgozók különböző szempontok szerint csoportosíthatók. A változási szempontok tizenkét félék lehetnek.

A PROGRAM FUTÁSÁHOZ SZÜKSÉGES ESZKÖZÖK:

1 DB **COMMODORE 64** SZÁMÍTÓGÉP
1 DB **VIC-1541** TIPUSÚ LEMEZEGYSÉG
1 DB **MÁTRIX** NYOMTATÓ
1 DB **KÉPERNYŐ** /TV KÉSZÜLÉK/

A program ára betanitással együtt 25000 Ft.

KERESKEDELMI SZERVEZÉSI INTÉZET

1134. BP. DÓZSA GYÖRGY ÚT 150.
Telefon: 202-650, 202-670
MARKETING OSZTÁLY

KERSZI

ZX-Spectrumra

REPÜLÉSSZIMULÁCIÓK PILÓTASZEMMEL

Régóta alkalmaznak szimulátorokat a repülőgépek személyzetének kiképzésére. Ezek élethűen megépített vezetőfülkék, amelyek az igazi repülés látványát, hangját és esetleg gyorsulásait is imitálják. Rajtuk „repülve" olyan vészhelyzetek élhetők át, amelyeket valódi repülőgépen túl kockázatos lenne gyakorolni.

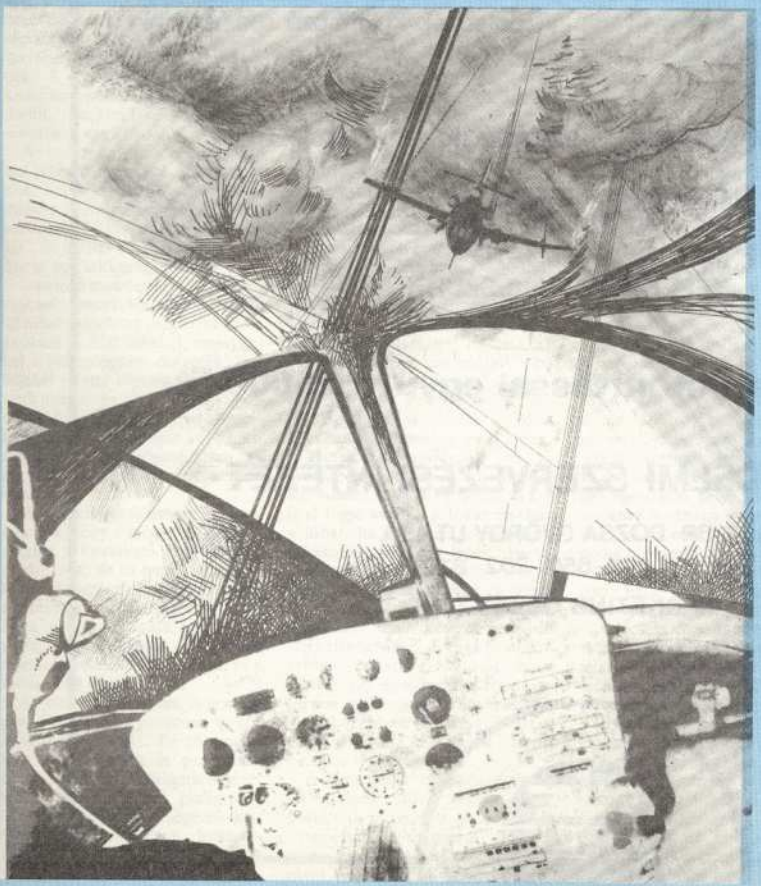
A személyi számítógépek fejlődésével a szimulációs programok kitörtek a kiképző központokból, és szélsebesen elterjedtek a kisgépeken. ZX-Spectrumon a következők a legismertebbek: Airliner, Delta Wing, Flight Simulation, Fighter Pilot, Nightflight.

A továbbiakban ezek közül a Flight Simulation-nel és a Fighter Pilot-tal foglalkozunk, mert ezek a legjobbak.

Kezdjük a leírások ismertetésével, hiszen sokan bizonyára ezek nélkül kapták a programot, vagy a nyelvi nehézségek akadályozzák a tanulmányozást.

Flight Simulation

Kis, egymotoros, 2–4 üléses túragépet vezetünk, ami főleg sétarepülésre készült, de néhány egyszerűbb műrepülő-figurát is elvisel. A képernyőn egyidőben láthatók a repülőgép műszerei *(1. ábra)* és az ablakon keresztül a környező táj, de átválthatunk a térképre is, amelyről gépünk helye és hozzávetőleges iránya is leolvasható.



Induláskor döntenünk kell, hogy milyen helyzetből akarjuk megkezdeni a repülést:
1. Valahol a levegőben repülve (in flight)
2. A leszállás végső fázisában (final approach)
3. A kifutópályáról felszállással (take off)

Ezután választhatunk, hogy akarunk-e szelet (Y) vagy sem (N). Végül megjelenik a műszerfal, és a kilátás. Kezdődhet a repülés.

A térképen két repülőtér található. Egy nagy, nemzetközi (MAIN), amelynek pályája pontosan kelet–nyugati irányban fekszik és hossza több, mint 2 km, és egy kisebb, sportrepter, amely 700 m hosszú, észak-déli betonnal. VIGYÁZAT! A térképen a pályák mérete torzított!

A pályák meghosszabbításában azok végétől 1–1 mérföldre a leszállás megkönnyítésére rádió-irányadók vannak. További három adó áll a három tó partján is. Mindig az az adó villog a térképen, amelyet a „B" gombbal kiválasztottunk, és amitől mért távolságot és irányt a navigációs műszerek mutatják.

A jobb felső sarokban 1000 láb magas hegyek vannak, amelyek ugyan a repülőgép ablakán át nem láthatók, de nekik lehet ütközni, ha 1000 lábnál alacsonyabban repülünk. Ha pedig 1000 és 2000 láb közötti magasságon szállunk fölöttük, a rádió-magasságmérő a hegyek tetejéről mért magasságot jelzi, ami persze pont ezer lábbal kevesebb, mint amit a másik magasságmérő mutat.

A biztonságos repüléshez tartsuk be az alábbi szabályokat:
- Ne ütközzünk földnek, hegynek vagy tónak!
- Ne szálljunk le behúzott futóval vagy 1000 láb/percnél nagyobb függőleges sebességgel!
- Ne guruljunk nagy sebességgel (gázzal) a pályán kívül!
- Ne repüljünk kisebb sebességgel, mint
  80 csomó behúzott fékszárnnyal,
  70 csomó félig nyitott fékszárnnyal,
  60 csomó teljesen nyitott fékszárnnyal!

Az utolsó szabály megszegése nem jelent automatikusan katasztrófát, csak „áteséshez" vezet, aminek következtében a gép egyik oldalra lebillen és süllyedni kezd. Ha van még elég magasság, gyors gázadással és kormányzással megmenthetjük a helyzetet.

Fighter Pilot

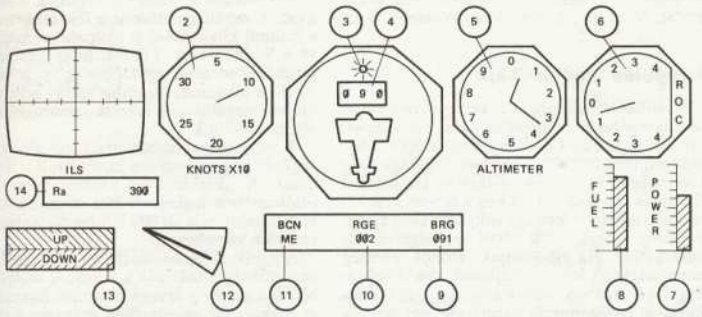
Az F 15-ös repülőgép adatai alapján készült. Teljesen műrepülhető, és a légi harcot is lehet gyakorolni rajta. Műszerfala a *2. ábrán* látható.

Leszállás (Landing Practice)

A gép 1700 láb magasan 6 mérfölddel a fő (BASE) pálya déli vége előtt repül a pálya irányába. A futómű kiengedve, kész a leszállásra. A leszállító műszer (ILS) vagy a fedélzeti számítógép jelzései alapján a magassági kormány, a fékszárny és a gáz segítségével szabályozható a sebesség és a süllyedés. Sikeres földetérés után a gázt teljesen le kell venni, majd a fékkel megállni.

Gyakorló repülés (Flying Training)

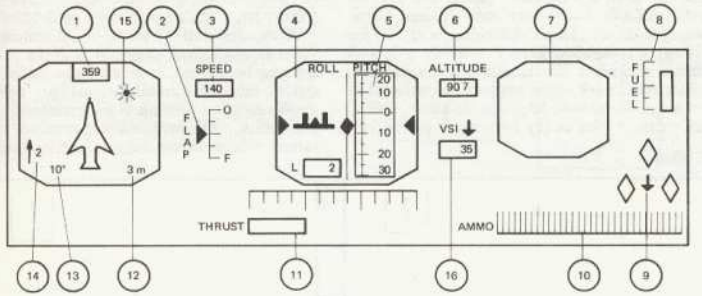
Kezdéskor a BASE pálya déli végén áll a gép, orral északnak. Felszálláshoz adjunk gázt. A felszálló sebesség elérésekor húzzuk hátra a botot. Úgy a legrövidebb a nekifutás, ha behúzott fékkel adjuk a gázt, és csak akkor engedjük fel, azaz kezdjük meg a nekifutást, ha elértük a teljes tolóerőt. Felemelkedés után sürgősen húzzuk be a futót, mielőtt elérjük a 300 csomós



1 „ILS" leszálló műszer
2 Sebességmérő-beosztás 10 csomókban (1 csomó = 1,85 km/ó)
3 Rádió-irányadó helyzete a géphez képest (villogó pont)
4 Repülési irány fokban
5 Magasságmérő
Kismutató: 1000 láb
Nagymutató: 100 láb (1 láb = 0,3 m)
6 „Variométer" a függőleges sebesség mérésére (beosztása 1000 láb/perc), felfelé: emelkedik, lefelé: süllyed
7 Motorteljesítmény
8 Üzemanyag-mennyiség
9 Iránya fokban
10 Távolsága mérföldben (1 mérföld = 1,85 km)
11 A beállított rádió-irányadó neve
12 Fékszárny ↑ behúzva, ↓ kibocsátva
13 Futómű ↑behúzva, ↓ kibocsátva
14 Rádió magasságmérő (csak 1000 láb alatt mér)

1. ábra. A Flight Simulator műszerfala

2. ábra. A Fighter Pilot műszerfala



1 Repülési irány fokban
2 Fékszárny ↑ behúzva, ↓ kibocsátva
3 Sebesség csomókban
4 Dőlési szög fokban
L: balra (left), R: jobbra (right)
5 Bólintási szög fokban
kék (sötét): felfelé, sárga (világos): lefelé
6 Magasság lábban
7 „ILS" leszálló műszer vagy fedélzeti számítógép
8 Üzemanyag-mennyiség
9 Futómű ↓ (zöld) kibocsátva, ↑ (piros) behúzva
10 Lőszermennyiség
11 Hajtómű-teljesítmény
12 Távolsága mérföldben
13 Rádióadó vagy ellenséges gép iránya fokban
14 Beállított rádió-irányadó jele
15 Rádióadó vagy ellenséges gép helyzete a repülőgéphez képest (villogó pont)
16 Függőleges sebesség (láb/másodperc)

sebességet. A felszálló sebességet csökkenthetjük a fékszárny kinyitásával is (140 csomóról 130-ra). Gurulás közben az oldalkormánnyal változtathatjuk a repülőgép irányát.

Légi harc gyakorlása (Air to Air Combat Practice)

A gép az ellenséggel azonos szinten, 2 mérföldet mögötte repül. Váltsunk át harci üzemmódra, és kapcsoljuk be a fedélzeti számítógépet. Így leolvashatjuk a másik gép irányát, távolságát és repülési magasságát. Az ellenséges gép 550 csomós sebességgel halad és nem viszonozza a tüzet, csak menekül. Ha már látja az ellenséget, próbálja befogni a célkeresztbe, és nyisson tüzet.

Légi harc (Air to Air Combat)

Ebben a fő üzemmódban meg kell oltalmaznia a négy repteret (BASE, TANGO, DELTA és ZULU). Kezdéskor a BASE pályáról felszállva a radar és a fedélzeti számítógép segítségével bemért ellenséget közelítse meg. Akkor láthatja meg, ha 1 mérföldnél közelebb van és a magasságuk különbsége kisebb 5000 lábnál. A saját gép sérüléseit a radar közepén levő kis repülőgép színváltozása jelzi. A negyedik találat végzetes. Ajánlatos a súlyosan sérült géppel kiválni a harcból. VIGYÁZAT! Ha a gép sérülése vagy a lőszer hiánya miatt menekülünk el, az ellenség úgy is eltalálhat egyszer, ha túl vagyunk az 1 mérföldön vagy az 5000 lábon. Az ellenség ilyenkor folytatja útját a célba vett repülőtér felé, ezzel időt nyerünk, hogy leszállhassunk valamelyik repülőtérre a gépet kijavítani vagy feltölteni.

Vakrepülés (Blind Landing)

Ez az üzemmód ködöt imitál. Egyáltalán nem látszik a horizont, és a talaj is csak 50 láb alatti magasságból. A műszerek és a térkép segítségével lehet navigálni.

Szél (Crosswind and Turbulence)

Hatására a gép elsodródik, és véletlenszerű szellőkések kibillentik addigi helyzetéből. Alaposan megnehezíti a repülést. Csak némi gyakorlat megszerzése után ajánlatos használni.

Az ellenfél ügyessége (Pilot Rating)

Változtatható a zöldfülűtől a zseniális „ász"-ig. A légi harcban nyilvánul meg:
– Milyen hamar veszi észre a közeledő vadászt?
– Milyen gyorsan fogja be a célkeresztbe az ön gépét?
– Milyen messziről lehet lelőni?

Az ön gépének tulajdonságát az ellenség ügyességi foka nem befolyásolja. Nem árt vigyázni az „ásszal", nagyon kemény ellenfél.

A botkormány fajtájának kiválasztása (Controls)

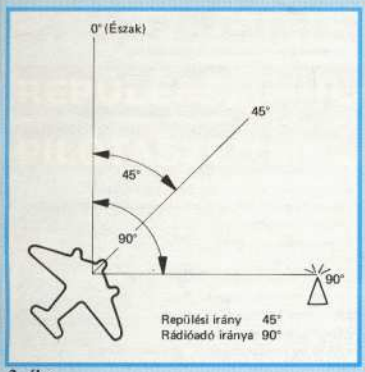
A biztonságos repülés szabályai:
– Ne ütközzünk földnek vagy hegynek!
– Ne szálljunk le behúzott futóval, 10 láb/percnél nagyobb függőleges sebességgel vagy 2–3 fokánál nagyobb oldalra dőléssel!
– Ne guruljunk 250 csomónál nagyobb sebességgel!
– Ne legyen kibocsátva a futómű 300 csomónál nagyobb sebességnél!
– Ne változtassuk a fékszárny állását, ha a sebesség nagyobb 472 csomónál!
– Ne repüljünk lassabban 130 csomónál csukott fékszárnnyal és 120 csomónál, ha a fékszárny teljesen nyitva van, mert „átesik" a gép!

A navigáció

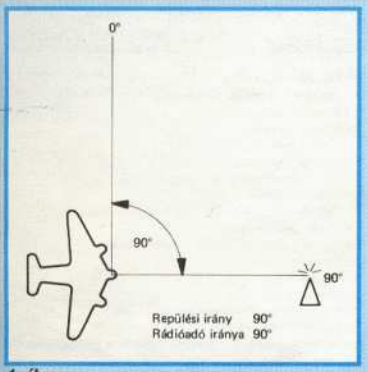
A repülésben az irányt nem égtájakkal fejezzük ki, hanem fokkal. A teljes kör 360 fok, a 0°-os irány az észak és jobbra haladva növekszik: észak 0°, kelet 90°, dél 180°, nyugat 270°.

Ilyen iránymeghatározással találkozunk a repülési irányt kijelző műszereknél (iránytű), a rádió-iránymérőkön és a kifutópályák fekvésének megadásánál.

A rádióadók iránya az az égtáj, amerre az adó a repülőgéptől esik (*3. ábra*). Ha a rádióadó felé akarunk repülni, addig kell fordul-



3. ábra



4. ábra

nunk, amíg a repülés iránya meg nem egyezik a rádióadó irányával *(4. ábra)*.

Bonyolultabb a helyzet, ha az sem mindegy, hogy milyen irányból repülünk az adó felé, mert például le akarunk szállni, és azt csak a pálya hosszában lehet. Nézzünk erre egy példát az *5. ábrán*. A kiinduló helyzetből nem 45°-os irányra fordulunk, hanem túl azon, az egyszerűség kedvéért 90°-ig. Ebben az irányban haladva azt tapasztaljuk, hogy a rádióadó irányát jelző szám egyre csökken. Amikor eléri a 0°-ot, akkor vagyunk a leszállópálya meghosszabbításában. Most már csak balra kell fordulnunk, hogy megkezdhessük a leszállást. A gyakorlatban persze nem akkor kezdjük a balra fordulást, amikor a pályát már kereszteztük, hanem egy kicsit előbb, hogy ne fussunk túl rajta fordulás közben.

Leszálláshoz a legnagyobb segítséget az ILS leszálló műszer adja. Tájékoztat, hogy milyen a gép eltérése az ideális helyzettől. A *6. ábrán* látható, hogy a műszer villogó pontja arrafelé mozdul el, amerre korrigálni kell az irányt. VIGYÁZAT! A pályához közeledve a műszer egyre érzékenyebben reagál az eltérésekre. Nehogy kis magasságban hirtelen erős süllyedésbe vigyük a gépet, mert pár méterrel feljebb vagyunk a kelleténél.

A fedélzeti számítógép nagyon pontos helymeghatározást tesz lehetővé, mivel a beállított rádióadóhoz viszonyított helyzetet lábban jelzi ki égtájak szerint (N = North = észak, S = South = dél, E = East = kelet, W = West = nyugat). Ez a műszer csak akkor mutat, ha az adó 6 mérföldnél közelebb van.

A repülés mechanikája

A repüléshez először is le kell győzni a súlyt, azaz levegőbe kell emelni az egész szerkezetet. Ezt repülőgépeknél a szárnyakon keletkező felhajtóerő teszi lehetővé, melynek nagysága függ a sebességtől és a szárny alakjától. Ha például vízszintes repüléskor csökken a sebesség, a felhajtóerő is kisebb lesz, így süllyedni kezd a gép. A szárny alakját a fékszárny segítségével változtathatjuk. Ha kibocsátjuk, azonos sebesség esetén nagyobb lesz a felhajtóerő, kisebb sebességgel is levegőben tudunk maradni. Ezt a tulajdonságát használjuk ki leszállásnál például, ha nem akarunk nagy sebességgel földet érni, és így túl sokat gurulni a megállásig.

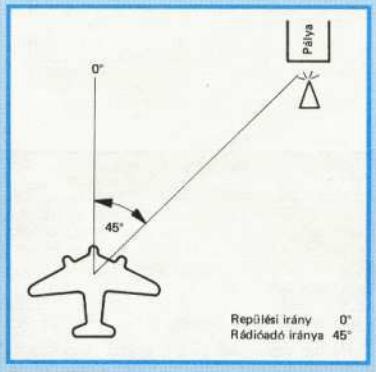
A légellenállás szintén a sebességtől, a repülőgép alakjától és a levegő sűrűségétől függ. Nagy sebesség eléréséhez nagy ellenállást kell leküzdeni, és ehhez persze nagy tolóerő szükséges, így lehet a sebességet gázzal szabályozni. VIGYÁZAT! A „Fighter"-ben egészen kis sebességeknél, az „átesés" közelében a szárny körül egyre erősebben kezd örvényleni a levegő, amitől megnő a légellenállás, így a sebesség csökkenését csak egyre nagyobb gázadással lehet megállítani, amit felgyorsulás követ, ezért le kell venni a gázt és így tovább. A gázzal való „fűrészelést" úgy lehet elkerülni, ha lassításkor 3–400 csomós sebességnél kinyitjuk a fékszárnyat. A repülőgép alakját a fékszárnyon kívül a futómű kibocsátása is változtatja, megnövelve a légellenállást. Fontos, hogy az ellenállás megnövekedésével egyidőben gázt is adjunk, mert ha elkésünk, megeshet, hogy nem tudjuk időben megállítani a sebesség vészes csökkenését, és „átesünk".

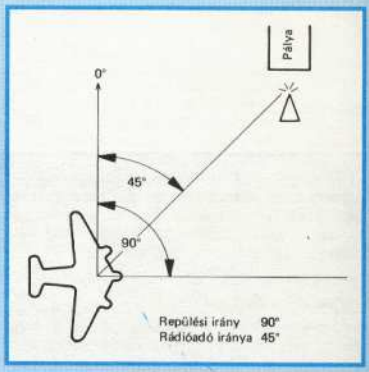
A nagy magasságban uralkodó alacsony levegősűrűség kellemesen lecsökkenti a légellenállást. A „Fighter"-nél például teljes gázzal földközelben legfeljebb 800 csomóra tudunk felgyorsulni, míg 60 000 láb magasságban 1440 csomó a végsebesség.

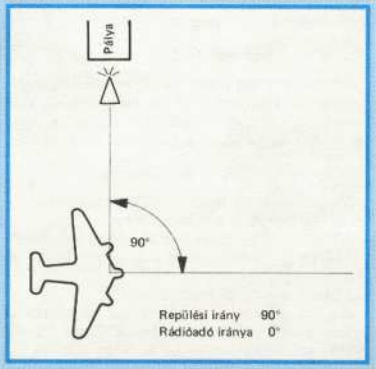
A légellenállás leküzdéséhez és az emelkedéshez szükséges tolóerőt a hajtómű szolgáltatja. Mivel a motor a levegő oxigénjét használja fel az üzemanyag elégetéséhez, magasan a ritkább levegőben lecsökken a teljesítménye. Vadászgépeken általában van utánégető. Ez a berendezés a teljes gáz ráadása után kapcsolódik be, ha továbbtoljuk a gázkart. Nagy mennyiségű üzemanyagot fecskendez a fúvócsőbe, megnövelve a tolóerőt, de még jobban a fogyasztást. A „Fighter"-en a tolóerő műszerének utolsó (piros) harmada az utánégető működését jelzi.

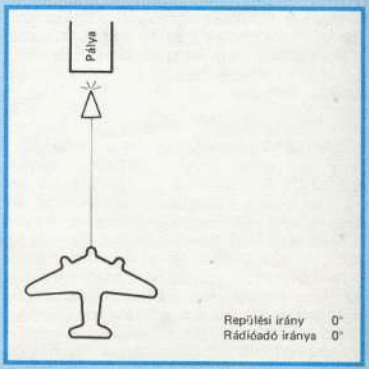
A repülés irányát fordulókkal tudjuk megváltoztatni. Ehhez elegendő a gépet a kívánt irányba bedönteni, és máris megkezdi az elfordulást, mégpedig annál gyorsabban, minél nagyobb szögben tértünk el a vízszintestől. Nagy bedöntésű, szűk fordulókban azonban tapasztalunk néhány furcsa dolgot. A fellépő centrifu-

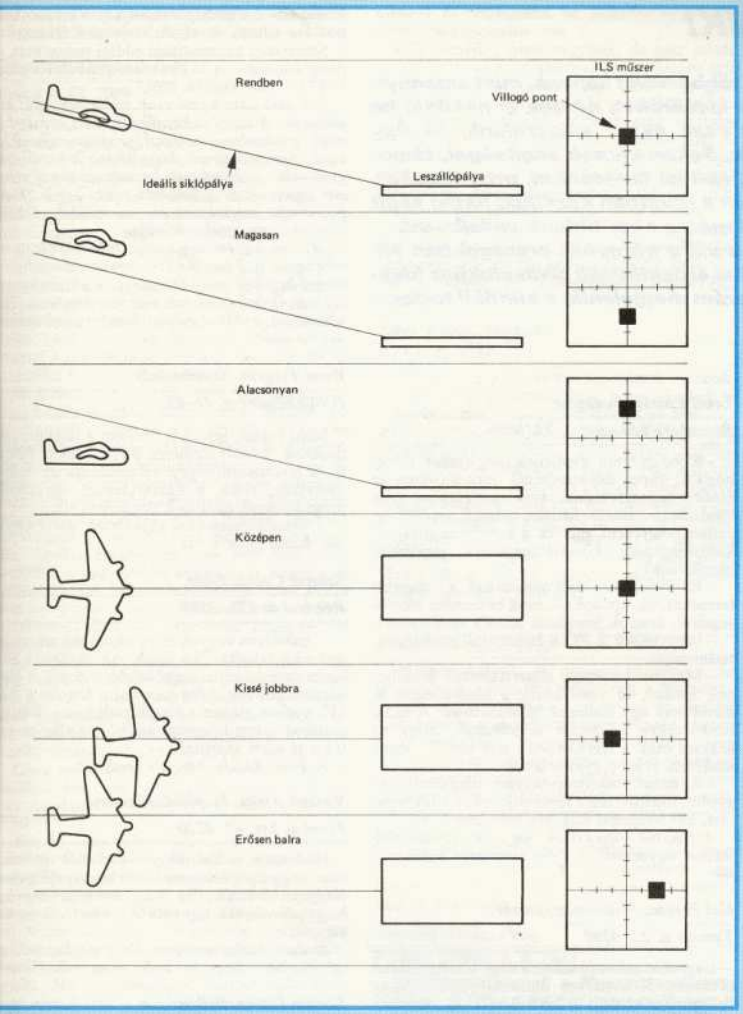
5. ábra











gális erő (vektoriálisan) hozzáadódik a repülőgép súlyához, és a gép vízszintes repülésből süllyedésbe kezd.

Igen nagy bedőntéseknél (45° fölött) észrevehetővé válik a kormányok hatásának felcserélődése: az oldalkormánnyal lehet az emelkedés vagy süllyedés mértékét szabályozni, míg a magassági kormány húzásával inkább a fordulót szükíthetjük.

A repülőgép-vezetés minden apró fogására természetesen már csak helyszűke miatt sem térhettünk ki, de célunk különben is csupán a programok használatának lehetővé tétele volt. Megfelelő érdeklődés esetén később magasabb szinten visszatérhetünk a témára, megtárgyalva olyan kérdéseket, mint a műrepülés, a légi harc, a navigációs számítások vagy a bonyolult időjárásban végrehajtott leszállás. Addig is mindenkinek sok jó leszállást! HÁY GYÖRGY

Program(ön)kritika

Több olvasónk is hiányolta az 1984/4. szám Százlábú rovatában, hogy a ZX-Spectrum fejléc nélküli LOAD SAVE rutinjához nem közöltük a kódot. Íme a decimális kód:

LOAD		
SCF	55	
LD(IX),B	221,33,N,N	(kezdőcím, előbb az alacsony, aztán a magas bájt)
LD(DE),C	17,N,N	(programhossz hasonlóan)
LD A,255	62,255	
CALL 1366	205,86,5	(ROM LD-bájt rutin)
RET	201	
SAVE		
LD(IX),NN	221,33,N,N	
LD(DE),C	17,NN	
LD A,255	62,255	
CALL 1218	205,194,4	
RET	201	

Így is lehet javítani az 1984/6. számban megjelent 1 k-s ZX81 Autóverseny programot:

7 LET A=A+INT(RND*3−1)+
+(A>0)−(A<18)
9 GOTO 10*(P=38 OR P=49)

A javítást Máté Gábor küldte be.

Lapunk 1984/5. számában adtuk közre a játékprogram-pályázat negyedik és ötödik helyezést elért programjait. Am a szerzők neveit sajnálatos módon felcseréltük. Helyesen: a „Jégcsapolvadás" című programot Pereczes Zsolt, az „Autózás" címűt pedig Tőkei Gyula készítette. A szerzőktől és az olvasóktól elnézést kérünk.

. ábra

FLIGHT SIMULATION

↑ (7)	Magassági kormány előrenyomása (süllyedés)	X	Oldalkormány jobbra (jobbra fordulás)
↓ (6)	Magassági kormány hátrahúzása (emelkedés)	P	Gázadás (nagyobb teljesítmény)
→ (8)	Csűrés jobbra (jobbra bedőlés)	O	Gáz levétele (kisebb teljesítmény)
← (5)	Csűrés balra (balra bedőlés)	F	Fékszárny kibocsátása
Z	Oldalkormány balra (balra fordulás)	D	Fékszárny behúzása
		G	Futómű kibocsátása-behúzása
		B	Rádió-irányadó kiválasztása
		M	Térkép-átkapcsoló

FIGHTER PILOT

↑ (7)	Magassági kormány előrenyomása (süllyedés)	A	Gázlevétel (kisebb teljesítmény)
↓ (6)	Magassági kormány hátrahúzása (emelkedés)	S	Fékszárny kibocsátása
→ (8)	Csűrés jobbra (jobbra bedőlés)	W	Fékszárny behúzása
← (5)	Csűrés balra (balra bedőlés)	U	Futómű kibocsátása-behúzása
CShift	Oldalkormány balra (balra fordulás)	N	Rádió-irányadó kiválasztása
Z	Oldalkormány jobbra (jobbra fordulás)	M	Térkép-átkapcsoló
Q	Gázadás (nagyobb teljesítmény)	B	Fék
		SShift	„ILS" leszálló műszer – fedélzeti számítógép-átkapcsoló
		C	Harci üzemmód átkapcsoló
		0	TŰZ (harci üzemmódban)
		H	Megáll a program
		J	Továbbindul a program

KEDVES OLVASÓINK!

Eljött az az idő, amikor már sokkal több levelet kapunk, mint amennyire válaszolni tudunk. Ezért kérem a levélírókat, nézzék el nekünk, ha hasznos tanácsaikra, javaslataikra csak akkor válaszolunk, ha úgy érezzük, hogy azokra választ várnak. Sokan kérnek segítséget, támogatást, például rajzokat, alkatrészt, építési tanácsokat, programokat. Ezeket a leveleket legtöbbször ebben a rovatban közöljük, hátha segít valaki; jelenlegi létszámunk mellett másra nem tudunk vállalkozni.

Meg szeretnénk szervezni, hogy a µM a környező országokban élő magyar anyanyelvű számítástechnikai érdeklődésű olvasóinkhoz hivatalosan is eljusson. Talán mire ez a szám megjelenik, a külföldi terjesztés gondja is megoldódik.

Lássuk a leveleket!

Malincsák János, Létavértes,

Nap u. 5. 4281

Írják meg, hova kell beküldeni azokat a programokat elbírálásra, amelyek az oktatási programajánlatban szerepelnek.

A programokat a Tudományszervezési és Informatikai Intézetbe (Bp., Egry József u. 1–9. 1111) kell küldeni. Küldhet oktatási programokat a µM-ba is, ha jók, megjelentetjük.

Fekete Sándor,

Románia

Szeretnék az Önök mikroklubjának levelező tagja lenni. Persze lehet, hogy erre nincs mód, de próbáljuk meg a lehetetlent.

Szeretnék magamnak csinálni egy személyi számítógépet, ami persze dokumentáció nélkül lehetetlen. Márpedig nekem semmiféle dokumentációm nincs, tehát Önöktől kérek segítséget.

Levelét átadtam a HCC elnökének, dr. Simonyi Endrének. Lemásoltam, és elküldtem az Építsünk számítógépet! sorozat eddig megjelent cikkeit, remélem, sikerül a számítógépet megépítenie.

Németh Róbert, Várpalota,

Táncsics u. 2. 8100

A nyár folyamán külföldi túránkról egy Laser 210 típusú 8 kbájtos számítógépet hoztunk, amely kiegészül egy 16 kbájtos memóriabővítővel. Az annyira áhított gépet tehát megkaptam, de sajnos nem nagyon tudom használni, mert nem tudok hozzájutni magyar fordítású gépkönyvhöz, illetve használatát magyarázó utasításhoz. Sajnos megfelelő fordítót nem találunk a környezetünkben. Mivel a Spectrum géphez több magyar nyelvű anyag is rendelkezésre áll, így próbáltuk ehhez hasonlóan működtetni, de sajnos nem egyezik meg a billentyű jelrendszerében. Ennek ellenére volt már sikerélményem is, a gépkönyvben lévő kisebb programmal. A beütött programot a gép beveszi, működteti, magnóra felveszi és a magnószalagról visszajátssza. De a gyári programos 16 kbájtos kazettát nem veszi be, és azt sem tudom, hogyan lehetne a grafikáján finomítani.

Példalevél külföldre utazóknak. Nagyon nagy kockázatot vállal, aki olyan számítógépet vásárol, amelyből kevés vagy egy sincs az országban. Nem tud használható leírást szerezni, nincs a gépnek szervize és nem tud programot cserélni sem. Sajnos!

Lehel László, Budapest,

Rosenberg házaspár u. 23. 1054

– Rendszeresen jelenjenek meg cikkek arról, hogy új gépek beszerzésénél mire ügyeljen az esetleg tapasztalatlan, szakismeretekkel nem rendelkező vásárló. Milyen nyugati gépeket érdemes beszerezni, melyek a kompatibilitás, alkalmazhatóság követelményeinek legjobban megfelelők?

– Rendszeresen tájékoztassanak a legújabb termékekről, típusokról, ezek beszerzési lehetőségeiről, árairól.

– Ismertessék a PC-k behozatali szabályait, tudnivalóit.

– Magyar kisgépek ismertetésénél közöljenek árakat, ez vonatkozik a hirdetésekre is, mivel csak úgy lennének informatívak. A hazai hirdetésekre különben is jellemző, hogy az árakról csak a legritkábban esik szó. Jó lenne szakítani ezzel a gyakorlattal.

– A hazai mikrogép-helyzet leírásánál nagyobb objektivitásra törekedjenek, s a takargatást, kendőzgetést hagyják más lapoknak.

Azt hittük, objektívak vagyunk, ezután még jobban vigyázunk. A többi javaslatot hasznosítjuk.

Gál Ferenc, Mosonmagyaróvár,

Tarcsai u. 22. 9200

Lapjukat számról számra nagy odafigyeléssel olvasom, 8. osztályos fiammal együtt. Nagy örömünkre szolgál a Tévé BASIC is, melyben az Önök érdemei elvitathatatlanok. Jó volna, ha lapjukban legalább néha helyet szorítanának a VC–20-szal kapcsolatos dolgoknak is. Ismerőseink közül egynek ZX81, ötnek VC–20 gépe van.

Az első VC–20 „programka" már megjelent az 1985/1. számban, de alig kapunk VC–20 programot. Nem küldene?

Póli Róbert, Foktő,

Duna u. 9. 6331

Ha a lap valóban laikusok, kezdők számára *is* készül, akkor szerintem a kezdők számára érdekes témákkal kellene foglalkoznia. Gondolok itt például a BASIC vagy bármely más programozási nyelv ismertetésére, több számon keresztül, méghozzá általános szinten is, mivel ilyen szakirodalmat nehéz találni.

A másik dolog, amit igencsak nehezményezek, hogy az 1984/3-as számban kiadott jelentkezési lapot a BASIC tanfolyamra kitöltöttem, postára adtam, de visszajelzés nem érkezett.

Szeretném hasznosítani eddigi tudásomat, és programjaimat is ki akarnám próbálni a µklubokban. Hol vannak ezek?

Sok javaslata közül csak szemelvényeket közölhetek. A lapot a kezdőknek is és a már komoly gyakorlatot szerzett programozóknak is írjuk. Arra számítunk, hogy a kezdők hamarosan gyakorlati szakemberek lesznek, és akkor szívesen lapozzák át a korábbi számokat is. Ezért beláthatja, nem lenne jó, ha minden cikkünk alapfokú ismereteket közölne.

Mire ez a szám megjelenik, mindenki biztosan megkapta a Tévé BASIC visszaigazolást és a vizsga helyéről szóló értesítést. A µklubok jegyzéke az Ötletben jelenik meg folyamatosan. Hétilap lévén, az Ötlet gyorsabban jut az olvasóhoz, mint a µM.

Rumi Frigyes, Szombathely,

Petőfi Sándor u. 41–43.

Mint sokan, így én is örültem a játékprogramoknak, de más területek is érdekelnek. Például: a mikroszámítógép alkalmazása az önálló nyelvtanulásban, a háztartásban, egy önálló programozott házi hi-fi stúdióban stb.

A témajavaslatokkal egyetértek, azért közöljük, hátha akad író is.

Szinyei Csaba, Gönc,

Rákóczi út 171. 3895

7. osztályos vagyok és az iskolában számítástechnika fakultációra járok. Az iskolának még nincs számítógépe, legközelebb Encsen, a gimnáziumban van. Meg szeretném tanulni a BASIC nyelvet, de nincs hozzá irodalmam. Kérem, ajánljon olyan könyvet, amiből az alapismereteket el lehet sajátítani.

A Tévé BASIC könyv a legújabb.

Vaszkó Attila, Hajdúböszörmény,

Fazekas krt. 47. 4220

Hallottam a Számítógép-kedvelők műsorában, hogy a jelentkezési lapok hamar elfogytak. Nagyon örülnék, ha meg tetszene engedni, hogy jelentkezési lap nélkül is részt vehessek a vizsgán.

Kedves Attila, remélem, Te is megkaptad már az értesítést, hogy hol tudsz vizsgázni. Várunk!

Kovács Gábor, Sellye,

Ifjúság u. 41. 7960

8. osztályos tanuló vagyok. Ebben az évben kezdtem foglalkozni a számítógéppel. Annyira megtetszett, hogy úgy döntöttem, hozzákötöm az életemet. Erről volna két kérdésem: lehet-e gimnáziumba jelentkeznem, és onnan olyan főiskolára mennem, ahol számítógépes foglalkozást tanulhatok? És ha igen, melyik ez a főiskola?

Örülök, hogy ilyen szép pályát választottál. Úgy tudom, hogy aki jó tanuló, azt biztosan felveszik a gimnáziumba, ezért kívánom, hogy nagyon jó legyen a bizonyítványod. Számítástechnikát ma már minden egyetemen és főiskolán tanítanak. Ha számítógép-tervezéssel szeretnél foglalkozni, akkor például a Kandó Kálmán Villamosipari Műszaki Főiskolát ajánlom, ha programozni, szervezni akarsz, akkor pedig a Pénzügyi és Számviteli Főiskolát. Azt hiszem, nem tévedek, hogy négy év múlva, amikor főiskolát

kell választanod, már bizonyára nagyobb választékot tudnék ajánlani. Részletes levelet küldtem.

Csizmadia Tamás, Komló,

Gorkij u. 13. 7300

A μM szakembergárdája fejtsen ki nagyobb aktivitást más lapokban is (természetesen ismeretterjesztési célokkal). Azaz lényegesen fokozni kell a nyomást a társadalomra. Ez pedig 30–40 ezres példányszámmal nem megy. Ha tehát a példányszám nem növelhető, akkor meg kell jelenni más, nagyobb lapokban.

A μM (az NJSZT-vel együtt) fejtsen ki nyomást az oktatásügy irányítóira, hogy a gimnáziumokban ne nagyon lehessen fakultációként pszichológiát, nevelést és hasonló tárgyakat felvenni, hanem igenis legyen több matematika, fizika, kémia, nyelvek, biológia. Ezeket a tárgyakat a továbbtanulásban a tanulók 80–90 százaléka fel tudja használni.

Megszüntették az ún. reál tagozatokat – nagy hiba volt. Most aztán hurrá, minden „elhumánosult". Tisztelet a kivételnek! Be kellene látnunk, hogy a magyar oktatásügy évszázadok óta erősen humán beállítottságú volt – ami tévedés volt. Amennyiben ezt a vonalat követjük, alighanem rövidesen sehol nem lesz Magyarország. Tévedés ne essék, nem a humán ismereteket sokallom, hanem a reál ismereteket keveslem. Amikor én érettségiztem, még nem csökkentették a művelődési anyagot, és mégis meg tudtuk tanulni. Ma is azt mondom – bár akkor nem így láttam –, hogy matematikából, fizikából többet kellett volna kapnunk. Pedig én reál tagozatra jártam.

Az eddigi oktatáspolitika eredménye az, hogy míg a bölcsészeti karokon a túljelentkezés eléri a tízszerest is, addig a műszaki egyetemeken alig másfélszeres, ha nem kevesebb. Be kell látni, hogy a technikai kultúrát háttérbe szorító oktatás a LEMARADÁS ISKOLÁJA!

Lapot nehéz szerkeszteni, egyelőre a legfőbb gondunk, hogy emeljük a μM színvonalát. Hogy más lapokban is írjunk, arra csak néha van erőnk.

Ami az oktatásügyet illeti, én is azt hiszem (ezt már néhány korábbi írásom is tanúsítja), hogy alapos reformra van szükség. Én azért nem szívesen esnék a ló másik oldalára, tehát óvakodnék attól, hogy kizárólag vagy főleg reál tárgyakat tanítsanak. Szerintem nagyobb szabadságot kellene adni a tanulóknak, hogy azt tanulják, ami érdekli őket, reál és humán tárgyakat együtt. Ma – véleményem szerint, ami lehet, hogy nem helyes – például a számítástechnikában olyan specialistákra lenne szükség, akik jól ismerik a szakterületet (például állattenyésztés, kórházszervezés, könyvtári rendszerek) és jó számítástechnikusok is, de az sem árt, ha a humán tárgyakban is műveltek. Azt hiszem, hogy az ilyen specialisták „előállítására" a mai merev felsőoktatási rendszer nem képes, de olyan javaslatokat kellene adnunk, amelyek ezt a problémát megoldják. Nem hiszem, hogy nem fogadnák szívesen.

Kovács László, Dunaújváros,

2408, pf. 24.

A televízióból értesültem arról, hogy az NJSZT figyelemmel kíséri az országban megalakuló mikroklubok működését. Barátaimmal létrehoztunk egy mikroklubot, amelynek az RCC (Rádióamatőr Computer Club) nevet adtuk. A klubhelyiség egy „feljavított" alagsori helyiség, amelyet közösen takarítottunk, tapéztáztunk és rendeztünk be. Jelenleg egy ZX81 típusú mikrogépünk van.

Pillanatnyilag öten vagyunk, de már jöttek érdeklődők, akik be akarnak lépni. Sajnos erre az egy gépre máris sokan vagyunk. Nagy gondunk, hogy a dunaújvárosi Munkás Művelődési Központ elzárkózott mindennemű segítségtől. Azt állították, hogy nincs helyiségük, és hogy ott soha nem lesz mikroklub. Nem egészen egy hónappal ezután említette meg egyik ismerősöm, hogy ott is beindítottak egy mikroklubot! Sajnos már belépni nem tudtam.

Én azt hiszem, hogy a Művelődési Központnak sem rossz, ha szakmai segítséget kapnak lelkes rádióamatőröktől. Ha kell, az NJSZT Fejér megyei vezetősége vagy a dunaújvárosi csoport biztosan segít.

Zsigó Bálint, Miskolc,

Iván u. 8. 3534

Olvastam a μM 1984/4. számának vezércikkét. Teljesen egyetértek azzal, hogy növeljék az egy évben megjelenő lapok számát. A példányszámot is emelni kellene, ugyanis néhány helyen nehezen lehet beszerezni. Nem tudtam megvenni az 1983-as számot. Kérném segítségüket, hogy ha van a szerkesztőségben még ebből, akkor küldjenek címemre egy példányt. Vagy ha nincs, mondják meg, hol lehetne beszerezni?

Mindenkinek üzenem, hogy az 1983-as szám egyáltalán nem kapható. Van néhány példány az 1984/3. és az 1985/1. számból, mással szerencsénkre nem szolgálhatunk.

Csikszenthy Csaba, Budapest,

Kádár u. 9–13/A. 1132

10 éves vagyok, és nagyon szeretném, ha felvennének a μklubba. Mi ennek a feltétele?

Még alig tudok BASIC nyelven, de már tudom kezelni a Honeywell és az Apple-Macintosh gépeket. Hamarosan kapok egy ZX–Spectrum személyi számítógépet. Apukám számítógép-matematikus, és Honeywell gépen dolgozik. Egy éve érdeklődöm a számítástechnika iránt, és gyakran bemegyek apukám munkahelyére.

A központi μklubba (V., Báthori u. 16.) és az NJSZT-be is a helyszínen lehet jelentkezni. Úgy látszik, az alma (Apple) nem esett messze a fájától, gratulálok.

Horváth Attila, Pécs,

Pollack Mihály u. 5. 7633

Sajnos nem vagyok a Magazin rendszeres olvasója, és főleg azért nem, mert eddig nem is nagyon tudtam a létezéséről. Gimnáziumba járok, és már másfél éve foglalkozom a számítástechnikával. Szeretnék valamilyen klubba belépni, mert sokszor akadnak problémáim, amelyekre elérhető szakkönyvekben nem találok választ, de sajnos nem tudok semmilyen klubról a közelben.

A μklubok listája először az Ötlet február 7-i számában jelent meg, a későbbi számokban pedig a folytatás.

Ifj. Füri Gábor, Szeghalom,

Május 1. út 2/1. 5520

Nagy számítógép-rajongó vagyok. Sikerült vennem egy VC–20-as számítógépet, amelyet igen megszerettem, jó volt rajta dolgozni. Múlt időben mondom, mert a gépem elromlott.

Mint már megtudtam, nem helyesen kezeltem. HT-hez voltam szokva, ezért az egyes elemek bekapcsolási sorrendjére nem fordítottam különösebb figyelmet. Ez volt a hiba. Később más géptulajdonosok mondták, hogy ez a géptípus igen gyengén van védve a külső hatások ellen, ezért a helytelen működtetés súlyos következményekkel járhat. Ez történt az én esetemben is. Gépem bebolondult. Körtünetei a következők: Első bekapcsoláskor a képernyő elsötétül, de a belső kép és a READY jelzés nem jelenik meg. Többszöri kapcsolgatás után – a kötelező szüneteket betartva – megjelenik a READY és a villogó kurzor. Ez azonban nem tart sokáig. A villogás megszűnik, és a gép érzéketlen lesz a billentyűzetre. Szóval bebolondul. Többszöri próbálkozás után megnyugszik: programozható lesz.

Sajnos már nem merem bekapcsolni a gépet, mert félek, hogy ezzel csak rontom a helyzetet. Ezért kérem Önt, hogy segítsen rajtam. Biztosan ismer olyan szakembert, aki meg tudná mondani, mi a gép baja, és hogy lehetne megjavítani. Kérem, juttassa el hozzá levelemet. Itt vidéken nincs olyan ember, akihez fordulhatnék ezzel a számomra igen súlyos gonddal.

Úgy tudom, hogy VC–20 szerviz nincs nálunk. Így csak a levél közzétételével tudok segíteni.

Tóth Sándor, Keszthely,

Ady Endre u. 22. 8360

A tanácsát szeretném kérni számítógépügyben. Előre kell bocsátanom, hogy a technikát kedvelő, de ebben az ügyben abszolút laikus vagyok. Szeretnék venni egy személyi számítógépet, amelyet tudnék használni elsősorban munkám során, másodsorban itthoni célokra. Egy 200 fős üdülőben raktárosként dolgozom. Hozzám tartozik kb. 1000 féle anyag, több csoportban. Elsősorban a munkámat szeretném megkönnyíteni a géppel, az itthoni felhasználás csak játék, szórakozás lenne.

A leveléből úgy értettem, hogy saját számítógépet akar venni, és ezt szeretné a munkahelyén is használni. Valószínűleg háttértárolós (mágneslemez) gépre lesz szüksége, ebből pedig csak a Commodore 64 kapható a magánforgalomban. Többet kellene a felhasználásról tudnom, hogy segíteni tudjak.

●

Egyetlen kérésem maradt, hogy olvasóink ne kíméljenek bennünket. Olvasói kapcsolat nélkül nem boldogulhat egyetlen szerkesztőség sem. A legközelebbi találkozásig megköszöni a leveleket:

KOVÁCS GYŐZŐ

Kabalapályázatunk véget ért!

Pályázatunkra igen sok ötletes rajz, vázlat érkezett, de helyhiány miatt csak egy részüket tudtuk közreadni. Épp ezért mondunk köszönetet mindazon játszótársainknak is, akik bár elküldték alkotásaikat, mégsem láthatták őket viszont lapunk hasábjain.

Kérjük kedves olvasóinkat, hogy nyílt levelezőlapon jelöljék meg azt a három kabalát, totemfigurát, amelyet a legszellemesebbnek találnak.

És most jöjjön a szavazás! Határidő: augusztus 15.

Címünk:
Mikroszámítógép Magazin Szerkesztősége
1027 Budapest II., Fő utca 68.
Pf. 451 Budapest 1372

Hírek, érdekességek

Bio-számítógép

Szilícium és más félvezetők helyett szerves molekulákból épül fel az a hagyományos elven működő számítógép, amelyen a japán Sharp cég szakemberei dolgoznak. A „Bio-morzsa" a molekulák nagyságrendjében működik, a méretcsökkenéssel a számítási folyamat is gyorsul. A gyártás során először rendkívül vékony szerves filmet készítenek, amelynek anyagából peptidek segítségével elektródot alakítanak ki. A ki- és bekapcsolás enzimek segítségével történik. Ebből ered a gép másik nagy előnye, ugyanis az enzimek vegyi reakciói jelentéktelen hőfejlődéssel járnak.

Igazgatók az iskolapadban

Az idén ősszel szakköri jelleggel Kecskemét valamennyi általános iskolájában bevezetik a számítástechnikát. Az akció első lépéseként az iskolaigazgatókat ültették a hazai gyártmányú iskolaszámítógépek mellé, nemcsak azzal a céllal, hogy a gépek kezelését megtanulják, hanem, hogy az alkalmazási lehetőségek megismerésével ösztönözzék őket az új technológia alkalmazására.

A nyolchetes tanfolyamot a kecskeméti Gépipari és Automatizálási Műszaki Főiskola szakemberei vezetik. Kecskemét 24 általános iskolájában több, mint 13 ezer diák tanul. Közülük először a hetedikesek és nyolcadikosok vesznek részt számítástechnikai képzésben, a hatodikosok pedig szakköri foglalkozásokon ismerkedhetnek a számítógéppel.

Utastájékoztatás Ferihegyen

Számítógéppel vezérelt lapozótáblák és nagyképernyős monitorok segítségével egyidejűleg 16 induló és 12 érkező járatról tájékozódhatnak majd az utasok a Ferihegyi repülőtér új forgalmi épületében. A rendszer működtetéséhez szükséges VT–20 alapú számítógépet, az azt kiegészítő különleges egységeket, valamint a szükséges programokat Székesfehérváron készítik.

Az új utastájékoztató rendszert március–április folyamán telepítették Ferihegyre, és a rendszertervező és kivitelező feladatokat ellátó VILATI irányításával augusztus végéig helyezik üzembe. A jó áttekinthetőség mellett előnye lesz a nagyfokú üzembiztonság és a megbízhatóság is. A számítógép ugyanis a menetrend adatait szolgáltató nemzetközi rendszer meghibásodása esetén önállóan is tudja irányítani az információs hálózatot.

Ferihegy korszerű tájékoztatási rendszerének elkészítésében több külföldi és hazai vállalat vesz részt.

Tv-Computer

Házi számítógép gyártását kezdte meg ez év második negyedévében a székesfehérvári Videoton Elektronikai Vállalat. A tavalyi tavaszi BNV-n már sikerrel bemutatott gép magyar ékezetes billentyűzettel készül és bármilyen típusú fekete-fehér és színes televízióhoz csatlakoztatható. Az utóbbi lehetőség már jelzi, hogy ez az első hazai gyártású, színkezelési képességgel is rendelkező házi számítógép. A gép háttértárolója háztartási kazettás magnetofon, de hajlékonylemezes tárolót is illesztenek hozzá.

A gyár még az idén kétezer darabot készít a Tv-Computerből, jövőre pedig már az igényeknek megfelelően tudja növelni a gyártást.

Elektronikus útvám

Hongkongban teljesen egyedülálló módszerrel kívánják mérsékelni az üzleti negyedek utcáin lassan elviselhetetlenné váló forgalmi dugókat és az ezzel járó fülsiketítő dudálást. Az elképzelések szerint a városállam fontosabb útvonalain csúcsforgalom idején autózók útvámot fizetnének. Ennek összegét úgy állapítanák meg, hogy a szóban forgó útvonalakon elektronikus érzékelőket helyeznének el, és a fölöttük áthaladó gépkocsikra vonatkozó információt egy központi számítógépbe továbbítanák. Az autótulajdonosoknak mintegy 50 dollárért meg kellene vásárolniuk azt a szintén elektronikus elven működő kódszám-kijelzőt, amelyet beépítenek majd a gépkocsijukba. Az autósok havonta megkapják a számlát, hogy a kijelölt területeken hol és hányszor haladtak keresztül, és ezért mit kell fizetniük.

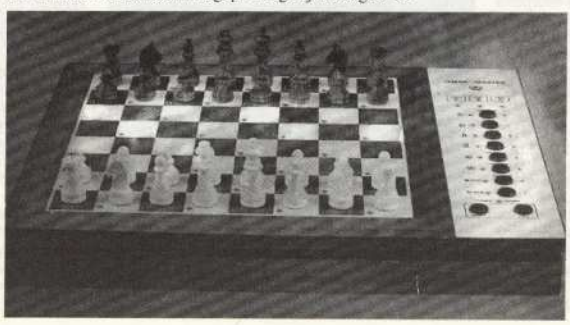
Az ellenzék véleménye szerint a kormány túlságosan nagy hatású eszközt kapna az autók mozgásának figyeléséhez, a megkezdett kísérleti telepítés vezetője azonban hangsúlyozta, hogy a rendőrség csupán akkor tekinthet bele az információkba, ha igen valószínű az autólopás vagy más bűncselekmény.

Az 5,3 milliós Hongkongban a lakosság mindössze 8 százaléka autótulajdonos, de az egy négyzetkilométerre jutó 270 autó feltehetően a világ legnagyobb autósűrűsége. Hivatalos becslések szerint az útvám következtében a csúcsidőszakban mintegy 10 százalékkal csökkenhet az autóforgalom.

A második sakkszámítógép

Az erfurti Mikroelektronikai Kombinátban készült el a szocialista országok első sakkszámítógépe, az SC 2, amely mellett most a közelmúltban bemutatták a második típust is: a Chess Mastert. A gép mikroelektronikai bázisát az első típussal megegyezően a Z80 mikroprocesszornak az NDK-ban gyártott, U880 elnevezésű funkcionális megfelelője képezi. A gép alapfunkciói az emberrel való játékon kívül lehetőséget biztosítanak a gépnek önmagával való játékára, két játékos számára döntőbírói szerep betöltésére. A játékos 8 erősségi fokozat közül választhat, természetesen színválasztási lehetőség is van. A gép járulékos szolgáltatásai: jó lépés tippelése, problémabevitel, a mérkőzés állásának megváltoztatása, feladványmegoldás: matt 1–4 lépésben. A Chess Master másodpercenként 15 lépésvariációt vizsgál meg, így „gondolkozási ideje" a választott erősségi fokozattól és a mérkőzés állásától függően 2 másodperc és 12 óra között változik.

Ennek a gépnek a prototípusa sikerrel szerepelt az 1983-ban Budapesten rendezett Sakkszámítógép Világbajnokságon is.



Kedves Olvasóink!

A jövőben örömmel adnánk helyet azoknak a híreknek is, amelyeket a vállalatok, szövetkezetek sajtóinformációként bocsátanának rendelkezésünkre. Amennyiben a közérdeklődésre számot tartó híreket folyamatosan beküldik szerkesztőségünkbe, bővebbé, színesebbé tehetjük hírrovatunkat. Máris élve e lehetőséggel, a Számítástechnika-Alkalmazási Vállalattól kapott információkból közlünk egy csokorral.

MÁTKA

Hamarosan eladó sorba kerül a MÁTKA, a Számítástechnika–Alkalmazási Vállalat Mezőgazdasági Általános Technológiai és Költségelemző Alkalmazás elnevezésű számítógépes programcsomagja. A szóba jöhető kérők, illetve vevők a termelőszövetkezetek és a kisebb állami gazdaságok lehetnek. A hazai fejlesztésű, valamint az IBM és vele kompatibilis személyi számítógépeken egyaránt alkalmazható program a számvitel adta lehetőségeknél részletesebben és naprakészen, a technológiai folyamatokkal összefüggésben elemzi az önköltségeket. A program figyelembe veszi az adott mezőgazdasági üzem sajátosságait. A kinyomtatott táblázatos jelentések mellett, a gazdálkodó egység vezetőinek módjukban áll a személyi számítógép képernyőjén keresztül bármikor információlódni valamennyi részleg pillanatnyi költséghelyzetéről.

A rendszer első eleme, a *Gépi üzem* alrendszer forgalmazását rö-

videsen megkezdi a SZÁMALK. Folyamatban van a *Növénytermesztés* és az *Állattartás* alrendszerek kidolgozása.

A MÁTKA rendszer bevezetése és üzemeltetése a mezőgazdaságban dolgozó vezetőktől és szervezőktől nem kíván több számítástechnikai ismeretet, mint amennyi a kézikönyvekből és néhány napos tanfolyam útján elsajátítható.

Számítógép-alkalmazók klubja

A közelmúltban újra megkezdte működését a szocialista országokban készült gépeket használó vállalatok, illetve szakemberek társadalmi szervezete, az ESZR/MSZR Alkalmazók Klubja. A Számítástechnika-Alkalmazási Vállalat Marketing Főosztályának támogatásával működő klub célja többek között, hogy lehetőséget nyújtson az információ- és tapasztalatcserére, konzultációt biztosítson a gyártók képviselőivel, segítse a kezdő szakembereket. A klub a Neumann János Számítógéptudományi Társaság Felhasználói Körének szervezeti egységeként Budapesten tevékenykedik.

ÁSZ bolt

Az ÁSZ az ÁPISZ Kereskedelmi Vállalat és a SZÁMALK közelmúltban megnyílt közös számítástechnikai szaküzlete. A Budafoki úti boltban közületi és magánvásárlók egyaránt megtalálhatják mindazokat a termékeket, amelyek a mikro- és kisszámítógépek napi üzemeltetéséhez szükségesek: mágneslemezeket, mágnesszalagokat, mágnesszalag-zárógyűrűket, festékkendőket, számítástechnikai szakkönyveket, különféle programokat, személyi számítógépes játékprogram-kazettákat. Az üzletben számítógép-árusítás is folyik. Erre a célra külön számítógép-bemutatótermet alakítottak ki. Az ÁSZ bolt árai több tekintetben igen alacsonyak. A tőkés import hajlékony mágneslemezek ára például mintegy 300 forinttal olcsóbb, mint a bizományi kereskedelemben.

TUDÁSPRÓBA

1. Melyik a legkisebb szám, amelyet sorozatban gyártott számológépben számábrázolási alapszámként használtak?

2. Van-e olyan szám, amelynél kisebb számmal mint számábrázolási alappal nem lehet digitális számológépet építeni?

3. Mit jelent a plug-kompatibilitás?

4. Mi a bug szó helyes kiejtése?

a) bug, b) bag, c) bág, d) badzs, e) bádzs.

5. Mit jelent a bistabil szó?

6. Létezik-e más, például monostabil és tristabil rendszer is?

7. Van-e olyan sajtótermék, amelynek nevében számítástechnikai szakkifejezés van?

8. Ki alkotta a kibernetika kifejezést?

a) A. M. Turing, b) N. Wiener, c) J. von Neumann.

9. Helyes-e a következő szómagyarázat? „Akkumulátor a számológépnek az a része (másképp regisztere), ahová például egy művelet eredménye vagy egy tárrekesz tartalma betölthető, áttölthető, és amelynek a tartalmához egy másik szám hozzáadható, abból kivonható vagy azzal összehasonlítható."

10. Mi az SBC?

11. Mi köze a programbeli ciklusnak a körhöz?

12. Mi a foolproof?

13. Az ALGOL vagy a FORTRAN nyelv jelent meg előbb?

14. Melyik nagyobb szám, a 9 alapú számábrázolási rendszerben a 6, vagy a 8 alapú számábrázolási rendszerben a 7?

15. Melyik volt az első, sorozatban gyártott modern (elektronikus, digitális, belső programozású stb.) számológép, amit hazánkban üzembehelyeztek?

16. Létezett-e olyan sorozatban gyártott számológép, amelynek egyetlen írásra és olvasásra alkalmas központi tára (ma már mindig perifériális egységként használt) dobtár (mágnesezhető réteggel bevont író és olvasó fejek előtt forgó henger) volt?

17. Mi az MTBF?

18. Igaz-e, hogy a lyukszalagokon kör alakú, a lyukkártyákon pedig téglalap alakú lyukasztást használnak?

19. Igaz-e, hogy „kétszeres pontossággal" (pontosabban: kétszeres szóhosszúsággal) dolgozva kétszer annyi értékes jegyű lesz a számolás végeredménye?

20. Mit jelent a CR?

A kérdésekre a választ a 48. oldalon közöljük.

Összeállította: TAKÁCSY ILDIKÓ

A programpályázat eredménye

Az 1985. március 22-én megrendezett II. Országos Számítástechnikai Verseny döntőjének színhelye ismét a szekszárdi Garay János Gimnázium és Óvónőképző Szakiskola volt. Az NJSZT, a Garay gimnázium és a μM szerkesztősége által meghirdetett pályázatra 50 program érkezett. Ezek közül a legjobbnak ítélt 16 került a döntőbe és szerepelt a nagy nyilvánosság előtt lezajlott versenyen.

A vetélkedőt Kovács Győző, az NJSZT főtitkára nyitotta meg, majd a Commodore 64 beszélőprogramja üdvözölte a megjelent vendégeket. A zsűri egyenként beszélgetett a versenyzőkkel a benevezett programjaikról. Ezután következett az értékelés.

A játékprogram kategóriában 10 program versenyzett. A fődíjat *Herendi Miklós* ABC–80 gépre írt, Kenya nevű programja kapta. A program elefántvadászatra invitál, amelyet élethűen modellez. Sok humorral kísért az elefántvadászt – egy olyan világban, ahol minden pénzbe kerül. Ebben a kategóriában a második helyezett *Gyuk Zsolt* ugyancsak ABC–80-ra készült, Kereskedő című programja lett. Ez egy üzletember munkájába enged betekintést. A játékos feladata sokrétű: számolnia kell a pénzügyi gondokkal és megfontoltan kell vásárolnia. A harmadik helyet *Tihor Miklós* programja, a Péter és Pál nyerte el. A program szinesen mutatja a két szomszédot, akik kertjeikben fákat ültetnek, és az azokra szálló madarakat gyűjtik.

Az oktatóprogram kategória döntőjébe 6 HT program került. A legjobbnak *Sándor Imre* Algoritmus nevű programját ítélték, amely a számok sorba rakásának technikáját mutatja be és magyarázza el. A második helyezett egy Otto-motort bemutató program lett. Készítője *Zsembery Péter*. A program szemléletesen ábrázolja a szerkezet felépítését és működését. A harmadik helyezett, *Márka Szabolcs* programja a gázrészecskék elrendeződését, két helyiségben található részecskék arányát és a különböző arányok valószínűségét is ábrázolja grafikonon.

A Híradástechnika Szövetkezet a legjobb HT gépre készült programért egy számológépet ajánlott fel. A díjat *Kovács Zoltán* nyerte, gépi kódú játékának kitűnő programozástechnikájáért.

A közönség díját *Tihor Miklós* és *Pámer Róbert* kapta.

A Tolna megyei Tanács különdíját *Sándor Imrének* és *Rátkai Istvánnak* ítélték.

ADOK – VESZEK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közleteknek gépelt soronként (60 karakter) 100,– Ft, magánszemélyeknek az első sor 50,– Ft, minden további sor 20,– Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

- **ÚJ COMMODORE VC–20** 18 ezer forintért eladó. Telefon: 06-80-50-991
- **VC–20 ALAPGÉPRE** Disassembler és Assembler program, továbbá játékprogramok eladók. Tóth Kornél, Bp. VIII., Leonardo da Vinci u. 29. fsz. 8.
- **ZX–SPECTRUMRA** programok kaphatók: havidíjasok bérelszámolása, címletezés, keresetszint-adózás, gmk-tagok jövedelemadózása. Füller Miklós, Bonyhád. pf. 66. 7151
- **KÖZVETÍTJÜK** személyi számítógépek adásvételét. Alacsony áron kínálunk megvételre Commodore VC–20 és C64, valamint ZX81 számítógépekhez tartozékokat és programokat. Kérje tájékoztatónkat! INTER GM. Kaposvár, Rákóczi tér 1. 7400
- **TI–99/4A-HOZ** vennék kiegészítőket és modulokat. Vári Dénes, Hódmezővásárhely, Kaptár u. 17.
- **VC–20-HOZ** magyar nyelvű programozási kézikönyv eladó. Ára 500,– Ft. Sági Endre, Cegléd bercel, Cseröi u. 10. 2737. Telefon 172-411/16-53
- **SZÍVESEN RÉSZT VENNÉK A CSEREAKCIÓBAN.** Van gépem, botkormánnyal és magnócsatlakozással. Egy cartridge kazettám is van, ALEN ADDITION nevű, összeadáson alapuló ügyességi játékkal. Tóthfalusy Péter, Bp. Klauzál tér 16. 1072
- **COMMODORE 64-RE** készült angol, amerikai, olasz játékprogramokat és utility programokat cserélne magyar felhasználókkal Roberto Banfi, Via Filippo Reina 18, 21047 Saronno (Varese), Olaszország. (Levelezés olasz vagy angol nyelven)
- **SZÖVEGFELVITELRE VÁLLALKOZÓKAT KERESÜNK** Commodore 64-re. Gép, lemezegység szükséges, nyomtató előny. Érdeklődni: 66-99-88
- **48 K-S ZX–SPECTRUMHOZ** sakk- és repülésszimulációs programot cserélnék más Spectrum játékprogramra. Kémeri József, Kecskemét, Botond út 2. C/IV/18. 6000
- **MEGVÉTELRE AJÁNLOM** VC–20-hoz és C64-hez kapcsolható Anitech számítógép-magnómat és játékkazettáimat (Commodore 64-re). Szabó József, Karcag, Kossuth tér 7–9. 5300
- **TIMEX SINCLAIR** gépemet oszcilloszkópra cserélném vagy elfogadható áron eladnám. Juhász János, Siófok, 8601 pf. 61.
- **ELCSERÉLNÉM VAGY ELADNÁM** VIC–20-as számítógépemet tartozékokkal. Cseréket ZX Spectrumot kérek. Murin Ákos. Telefon: 649-885

BASIC bajnokság

(Budapest, 1984. Számítástechnikai Koordinációs Intézet, 139 oldal. Ára: 60,– Ft.)

A Számítástechnikai Koordinációs Intézet és leányvállalatai, a Sci–L és a Scitel az utóbbi években a Budapesti Nemzetközi Vásáron számítógépes BASIC-vetélkedőt rendeztek, amelyek sikere arra ösztönözte a szervezőket, hogy a verseny feladatait könyv formájában is kiadják. Így az olvasók az egyes feladatokat hosszabb ideig tanulmányozhatják, és a megoldások kiválasztása közben programozási ismereteiket is bővíthetik. A vetélkedőre eredetileg csak MO8X gépre írt programokat a könyv szerkesztője a Magyarországon jelenleg legelterjedtebb mikroszámítógépekre is átíratta. Ezáltal lehetővé válik a HT–1080Z, a Commodore 64 és a Sinclair Spectrum gépeken egyaránt megoldott feladatok összehasonlítása.

Hice, G. F. – Turner, W. S. – Cashwell, L. F.: Számítógépes rendszerek fejlesztésének módszertana

(Budapest, 1983. Műszaki Könyvkiadó, 433 oldal. Ára: 78,– Ft.)

Végre megindul a szervezetten és ellenőrzötten végzett számítástechnikai munka is hazánkban. Végre! Sok-sok éves szónoklás és „fogjuk meg és vigyétek!" stílusú konferenciásdi, irányelvesdi, bizottságosdi és határozatosdi után.

Egyetlen könyvtől? Hiszen egy fecske nem csinál nyarat. Csinálni ugyan nem csinál, ez igaz. De annyit egyetlen fecske is el tud érni, hogy eszünkbe juttatja a nyarat. Ez a könyv azonban sokkal többet ért el annál, hogy eszünkbe juttatta, hogy van gondosan szervezett és végigvezetett számítástechnikai munka is. Azt is elérte, hogy e munka részleteinek és fogásainak – ha nem is minden részletében vitán felül álló, de – nagyon gazdag, a gyakorlatban vizsgázott készlete is összegyűjtve, rendezetten a kezünkbe került.

Ennek következtében viszont elkerülhetetlenné válik, hogy meginduljon a számítástechnikai munka értékelemzése, ha nem is rendszeresen és intézményesen, de ösztönösen és szubjektíven. Ma nálunk ez a óriási eredménynek számít. Megjegyezzük, hogy komoly értékelemzési alapmunka évtizedek óta nem látott napvilágot Magyarországon.

Egy, a számítástechnikai tevékenység lélektani vonatkozásaival foglalkozó, nem érdektelen mű néhány évvel ezelőtt már megjelent hazánkban. (Jelenleg azonban már ez sem kapható.) Taylor 1911-ben kiadott, korszakformáló, világhírű munkaszervezési könyvét 1983-ban (72 év után!) lefordították magyarra. Sokat tanulhat belőle az is, aki a számítástechnika szervezésével foglalkozik.

Ezzel aztán ki is merült a számítástechnika-alkalmazás módszertana iránt érdeklődőket segíteni tudó, jelentősebb magyar nyelvű művek listája. Egy kiemelkedő munkát még meg kell említenünk, Fuori, D'Arco és Orilia „Számítógépek kezelése" című könyvét, amelyben spontán érvényesül a munkaszervezési szemléletmód. (E kiváló munka megjelentetése is a Műszaki Könyvkiadó érdeme.

Hice, Turner és Cashwell munkája jellegében rokon Fuori, D'Arco és Orilia könyvével, ugyanis ez is arra törekszik, hogy a tárgyalt terület minden lényeges elemét érintse. Aki elkezdi olvasni, nagyon hamar azt veszi észre, hogy benne van egy légkörben. Hogy forr, él körülötte a munka, amely nemesít, fejleszt, amelynek sikere nekünk is örömöt szerez. És ez a munka abbahagyhatatlanul érdekes, magunkat és a közösséget is gazdagabbá teszi. De csak ha olyan szellemben végezzük, ahogyan e könyv tárgyalja. Mert a munka lehet puszta kín is, de lehet felemelő, felszabadító tevékenység is. Hogy mikor melyik, azt a tevékenységnek az igazsághoz, a rendhez, az értékekhez való viszonya dönti el.

Ilyen szempontból minden bizonnyal vízválasztó a tevékenység értelmessége. Ehhez viszont elengedhetetlen annak minden – nemcsak pénzügyi, hanem humán – igényt kielégítő szervezettsége. Ennek eléréséhez segít hozzá ez a hazai téren úttörő könyv, melynek páratlanul gazdag anyaga ismeretét ma már nem nélkülözheti egyetlen olyan számítástechnikus sem, aki ad valamit magára és munkájára.

Könyvbírálat lett volna a feladatunk. Most azonban engedtessék meg, hogy ne legyünk ünneprontók a részletes elemzések fontos, de a jelen születésnapi hangulatba nem illő elvégzésével, és csak néhány jótanácsot adjunk arra vonatkozóan, hogy a számítástechnika-módszertan e szerencsésen választott import csecsemője harmonikusan és egészségesen fejlődhessen tovább, és kiheverhesse a világra segítés során elszenvedett kisebb-nagyobb torzulásokat.

Módszertani munkáról, mégpedig sokakra ható módszertani munkáról lévén szó, sokkal jobban kell ügyelni a szakkifejezések pontosságára és jóságára. (Különbség van például tábla és táblázat között, – hogy csak egyet említsünk, azt is az enyhébbek közül.) Ugyanilyen okokból célszerű a magyarra is lefordított munkákat a magyar címmel és adatokkal együtt külön megemlíteni.

Nemcsak olyan számológépes rendszerek vannak, amilyenekkel ez a nagyon értékes könyv foglalkozik. Vannak például műszaki–tudományos, gazdaságmatematikai, számológépes technológiai folyamatirányítási, robottechnikai, NC és CNC rendszerek is. Ezek létesítésének pedig megvannak a jellegzetes módszertani sajátságai, amelyeknek legalább vázlatos megemlítése – kiegészítés vagy függelék formájában – teljesebbé tenné az ilyen munkát, amelynek a teljességre törekvés az egyik fő célja.

Végül pedig vannak és lesznek egyre nagyobb számban és fontosságban személyi gépekre készített rendszerek (rendszerecskék) is. Ezek kifejlesztésének, üzemeltetésének módszertanát még nem írták meg. Ezt a feladatot Hice, Turner és Cashwell könyve után könnyebb is és nehezebb is lesz elvégezni. Könnyebb, mert van már miből tanulni. Nehezebb, mert azt a munkát majd ehhez a könyvhöz fogják hasonlítani.

POGÁNY CSABA

James W. Coffron: Mikroprocesszoros rendszerek gyakorlati hibakeresése

(Budapest, 1984. Műszaki Könyvkiadó, 245 oldal. Ára: 80,– Ft.)

Az amerikai szerző először a mikroprocesszorok felépítésével foglalkozik, majd a CMS hardverelemző alkalmazását írja le a legismertebb mikroprocesszorok vizsgálatánál. Nagyon ötletes módszereket mutat be, de az egyszerű vizsgálati eljárásokat is ismerteti. Hasznos tanácsokat ad a hibakeresés menetére, a vizsgáló ROM programjára vonatkozóan. Végül a TRS–80 személyi számítógépen konkrét hibakereséseket mutat be.

A könyv eléri célját, melyet a szerző így fogalmaz meg: „Célom az volt, hogy olyan könyvet írjak, amely világosan megmutatja, hogyan fogjunk hozzá a nyolcbites mikroprocesszorrendszerek gyakorlati hibakeresésához. Gondosan kifejtett, logikusan felépített, érthető és bőségesen illusztrált útmutatót igyekeztem adni olyan személyek számára, akiknek szándéka ilyen jellegű munkát végezni. Ha törekvésemet siker koronázza, az olvasó a könyv elolvasása után minden félelem és habozás nélkül képes lesz a nyolcbites mikroprocesszor-rendszerekben a hibát megkeresni."

Egyéb figyelemreméltó kiadványok

Commodore 64. Software alkalmazói segédlet (Budapest, 1984. Ipari Informatikai Központ, 416 oldal. Ára: 360,– Ft.)

A szocialista országok mikro- és miniszámítógépei. Összeállította: dr. Broczkó Péter (Budapest, 1984. Központi Statisztikai Hivatal, 37 oldal. Ára: 48,– Ft.)

VÁLASZOK A TUDÁSPRÓBA KÉRDÉSEIRE

1. Tudomásunk szerint a *mínusz kettő*. A lengyel UMC–1 típusjelű gép mínusz kettő alapú számábrázolási rendszerrel dolgozott. Ilyen gép hazánkban is működött az 1960-as években.

2. Nincs, mert például bármelyik negatív egész szám is használható számábrázolási rendszer alapszámaként. Ha pedig kikötjük az alapszám pozitivitását, akkor például kettőnek bármely negatív egész kitevőjű hatványa lehet alapszám, és 2 negatív egész kitevőjű hatványai között végtelen sok olyan van, amely kisebb bármilyen rögzített pozitív számnál.

3. Két egység plug-kompatibilis egymással, ha azonos feladatok elvégzésére alkalmasak, és még más egységekhez való csatlakoztatási lehetőségeik is megegyeznek. Tehát például az egyik csatlakozóját kihúzva és bedugva a másikét, a berendezésegyüttes ugyanúgy működtethető ezzel az egységgel, mint az előbbivel. A *plug* angol szó (ejtsd: plág) itt dugaszolót, csatlakozót jelent.

4. A c). A bogár, poloska (és átvitt értelemben) programhiba jelentésű angol szó helyes kiejtése bág, rövid á-val.

5. Egy rendszer bistabil, ha 2 stabil állapota van. Ilyen például a kétállású háztartási elektromos billenő kapcsoló vagy a torusz (mentőöv), vagy a felfújt gépkocsitömlő sík lapon, a nehézségi erőtérben.

6. Létezik. *Monostabil* például az inga, a kelj-feljancsi játék, a csengőgomb. *Bistabil* például a lencse, a lapos ellipszoid, a diszkosz. *Tristabil* például a forgástengelyén átmenő síkkal elvágott fél diszkosz. A sorozat azonban korlátlanul folytatható. *Tetrastabil* például a szabályos háromszög alapú egyenes güla. *Pentastabil* például a szabályos négyszög alapú egyenes güla és a szabályos háromszög alapú egyenes hasáb. *Hexastabil* például a szabályos ötszög alapú egyenes güla és a szabályos négyszög alapú egyenes hasáb, például a kocka. És így tovább.

7. Több is. A BIT, a BYTE, a CHIP szakkifejezés például mind ismert folyóiratcím is.

8. Egyik sem. A szó mindegyiküknél korábbi eredetű.

9. Nem. Főleg azért nem, mert így a tár minden rekesze akkumulátor lenne.

10. *Single Board Computer* kezdőbetűiből alkotott szó, egyetlen nyomtatott áramkörі kártyán elférő, így gyártott (és így forgalmazott) számológép. (Single = (itt) egyetlen, Board = (itt) áramkörі kártya, Computer = = számológép).

11. A program operátorainak kapcsolatábrája az ún. „folyamatábra" gráfjában ún. (irányított) kör, idegen szóval ciklus van. Ez a gráfbeli ciklus és a geometriai kör topológiailag homeomorf, azaz kölcsönösen egyértelmű és folytonos transzformációval egymásba alakítható.

12. Nagymértékben elronthatatlan, eltévesztehetlen; különlegesen biztonságos, kezelési hibáktól védett. Olyan berendezés, illetve eljárás jelzője, amit szinte még a bolondok sem tudnak elrontani, illetve másképpen végezni, mint ahogyan azt tervezték. (Fool = bolond, proof = (itt) -álló, szó szerint bolondálló, mint például vízálló, tűzálló, kopásálló.)

13. A FORTRAN.

14. Az utóbbi.

15. Az M–3 típusjelű, szovjet gyártású, később hazai szakemberek által továbbfejlesztett gép.

16. Igen. Magyarországon például a szovjet gyártású Ural–1 típusjelű gép.

17. A *Mean Time Between Failures* (Faults) kezdőbetűiből alkotott szó. A hibátlan rendelkezésre állási időszakok hosszának átlagos értéke. A meghibásodás, a rendelkezésre állás egyik fontos jellemzője. Jelentése: szomszédos hibák közötti átlagidő.

18. Általában igen, de van (volt) lyukszalagos rendszer, amely téglalap alakú és lyukkártyás rendszer, amely kör alakú lyukasztást használt.

19. Nem. Még az sem igaz, hogy minden esetben pontosabb lesz az így kapott eredmény, mint az „egyszeres pontossággal" végzett számolásé. Sőt pontatlanabb is lehet.

20. A *Carriage Return* kezdőbetűiből alkotott rövidítés. Szó szerinti jelentése „kocsi visszа". A kocsis elektromos írógépeknél a sor elejére állást vezérlő billentyű felirata. Majdnem mindig a *következő* sor elejére állást eredményezi, miután a papír egy sorral való továbbítását vezérlő, más szóval „soremelést" eredményező billentyű az utóbbi időkben kihalt. A képernyős megjelenítés esetében az *íráshelymutatónak* (a cursornak) a következő sor elejére állását vezérlő és (sor)lezárásról is tudósító billentyű felirata.

SOFTINVEST

SZOFTVER KERESKEDELMI ÉS FEJLESZTÉSI BETÉTI TÁRSULÁS 1525. BUDAPEST PF. 51. TEL.: 358-530/866.

Alapítók:

Központi Statisztikai Hivatal
Ipari Minisztérium
Művelődési Minisztérium
COMPORGAN
ÉGSZI
FÜTI
KSH SZÜV
OKISZ SZSZV.
SZÁMALK
SYSTEM

Minden piacképes SZOFTVER termék forgalmazásában, vagy ötlet megvalósításában partnerek vagyunk

– Közös kockázatú fejlesztések
– forgalmazói tevékenység (késztermék bizományosi értékesítések)
– ügynöki tevékenység (eladó–vevő kapcsolat létrehozása)
– fővállalkozói funkciók ellátása,
– szakmai tanácsadás
– információszolgáltatás

Azonnali szállítással ajánljuk a következő szoftvereket:

1. IBM PC–XT és minden kompatíbilis mikroszámítógépre, ezen belül kiemelten PROPER 16-ra, VT–16-ra:
   – Főkönyvi és folyószámla könyvelés,
   – Készletgazdálkodás,
   – Bér- és munkaügyi rendszer.

   Ezen géptípusra továbbá vállaljuk nagyobb projektek menedzselését, esetleg fővállalkozási formában.

2. MO8X, VT–20/A és IV, TZ–80, MULTICENTER, COMMODORE 64 gépekre ajánljuk a *CSB–807* általánosan használható szoftvert, amelynek segítségével *programozói munka nélkül* létrehozhatók, karbantarthatók és a legváltozatosabb szempontok szerint lekérdezhetők egyszerű és bonyolult nyilvántartások.
3. COMMODORE 610, 720 gépre:
   – Főkönyvi és folyószámla-könyvelés.
4. COMMODORE 64 gépre:
   – Széleskörűen használható vendéglátóipari és egészségügyi programcsomagok.

A SOFTINVEST minden általa forgalmazott szoftvert értékesítés előtt szakértőkkel minőségileg bevizsgáltat.

Minden szoftver értékesítésénél szolgáltatást is ajánlunk, nevezetesen tanácsadást, ha szükséges, szervezői kapacitás biztosítását.

SOFTINVEST SOFTINVEST SOFTINVEST SOFTINVEST SOFTINVEST SOFTINVEST SOFTINVEST

SZOFTVER BÖRZE

a MIGÉRT Bemutatótermében!

COMMODORE 64 személyi számítógépre készült szoftverek
széles választékával várjuk Kedves Ügyfeleinket.

Általános célú felhasználói programok
Feladat-orientált felhasználói programok
Oktató és Játékprogramok
Compilerek és fejlesztést segítő programok
Rendszer-programok
Utility segédprogramok
Adatátviteli programok (emulátorok)

A kiválasztott programok helyben kipróbálhatók és megvásárolhatók!

ÚJDONSÁG! ÚJDONSÁG! ÚJDONSÁG! ÚJDONSÁG!

Commodore 64 felhasználók figyelmébe ajánljuk az alábbi
hardver-kiegészítőket:

– TEDIFACE INTELLIGENS TELEFON MANAGER (I.T.M.)
– CE-LUX I. és II. típusú Fényceruza
– ROMOM K 6311. mátrixnyomtató + interface

A berendezések megtekinthetők
és megrendelhetők:

MIGÉRT

Bp. VIII. Rákóczi út 57/a.

Műszer és Irodagépértékesítő Vállalat
Bemutatótermében